



ANÀLISI, DISSENY I IMPLEMENTACIÓ D'UNA EINA  
DE SIMULACIÓ PER A L'ESTUDI DE LA  
INTEROPERABILITAT MITJANÇANT CONVERSIÓ  
D'ATRIBUTS EN ENTORNS MULTI-DOMINI

Memòria del projecte de final de carrera corresponent  
als estudis d'Enginyeria Superior en Informàtica pre-  
sentat per Fco. Javier Llorente Palacio i dirigit per  
Carles Martínez García.

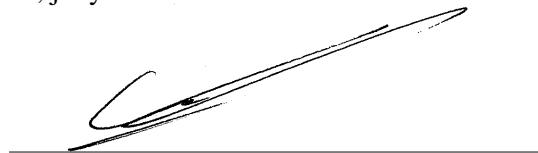
Bellaterra, juny de 2009

El firmant, Carles Martínez García , professor del Departament d'Enginyeria de la Informació i de les Comunicacions de la Universitat Autònoma de Barcelona

CERTIFICA:

Que la present memòria ha sigut realitzada sota la seva direcció per Fco. Javier Llorente Palacio

Bellaterra, juny de 2009

A handwritten signature in black ink, consisting of a series of loops and a long horizontal stroke, positioned above a horizontal line.

Firmat: Carles Martínez García

*A todas las pequeñas estrellas que iluminan el camino*



# Agraïments

Amb aquestes línies m'agradaria donar les gràcies als meus pares per l'oportunitat que m'han donat, als qui dec el fet de poder estar escrivint aquests agraïments ara mateix. Agraïxo també molt, l'oportunitat que m'ha donat el departament del DEIC. Dono les gràcies a cada un dels membres del departament i en especial al meu coordinador Carles Martínez García per la seva dedicació i paciència infinita.



# Índex

<b>1</b>	<b>Introducció</b>	<b>1</b>
<b>2</b>	<b>Fonaments</b>	<b>7</b>
2.1	Conceptes . . . . .	7
2.1.1	Control d'accés . . . . .	8
2.1.2	Interoperabilitat . . . . .	9
2.2	Elements del l'escenari . . . . .	11
2.2.1	Dominis . . . . .	11
2.2.2	Funcions i Escenaris de conversió . . . . .	12
2.3	Tecnologia a utilitzar . . . . .	15
2.3.1	Java i Swing . . . . .	15
2.3.2	XML . . . . .	15
<b>3</b>	<b>Anàlisi</b>	<b>17</b>
3.1	Anàlisi de viabilitat . . . . .	17
3.1.1	Viabilitat Tècnica . . . . .	18
3.1.2	Viabilitat Econòmica . . . . .	18
3.1.3	Viabilitat Legal . . . . .	19
3.1.4	Viabilitat Temporal . . . . .	19
<b>4</b>	<b>Disseny i implementació</b>	<b>21</b>
4.1	Disseny mòduls principals . . . . .	21
4.1.1	Mòdul: Escenari . . . . .	22
4.1.2	Mòdul: Dominis . . . . .	23

4.1.3	Mòdul: Atributs . . . . .	24
4.1.4	Mòduls: Funcions de conversió . . . . .	25
4.2	Disseny mòduls funcionals . . . . .	29
4.2.1	Mòdul: Xarxes públiques . . . . .	30
4.2.2	Mòdul: Xarxes privades . . . . .	33
4.3	Let's draw: MainDrawClass . . . . .	35
4.3.1	Entorn de dibuix . . . . .	35
<b>5</b>	<b>Proves i rendiment</b>	<b>39</b>
5.1	Resultats . . . . .	39
5.1.1	Resultats - Xarxes Públiques . . . . .	40
5.1.2	Valoracions resultats xarxes públiques . . . . .	41
5.1.3	Resultats - Xarxes privades sense cicles . . . . .	43
5.1.4	Valoracions resultats xarxes privades sense cicles . . . . .	44
5.1.5	Resultats - Xarxes privades amb cicles . . . . .	45
5.1.6	Valoracions resultats xarxes privades amb cicles . . . . .	46
5.2	Valoració resultats . . . . .	49
5.2.1	Comparació entre xarxes privades i xarxes públiques . . . . .	49
5.2.2	Valoració rendiment i resultats obtinguts . . . . .	51
5.3	Valoració de la configuració per xarxes privades . . . . .	52
5.3.1	Xarxes privades amb TTL i timeout . . . . .	52
<b>6</b>	<b>Conclusions</b>	<b>55</b>
	<b>Bibliografia</b>	<b>61</b>



# Índex de figures

3.1	Diagrama de Gantt . . . . .	20
4.1	Conversió d'atributs - Escenari . . . . .	27
4.2	Conversió d'atributs - Selecció valors mínims en camins . . . . .	27
4.3	Conversió d'atributs - Selecció valor màxim dels camins . . . . .	27
4.4	Conversió d'atributs - Selecció valors mínims en camins . . . . .	28
4.5	Conversió d'atributs - Selecció valor màxim dels camins . . . . .	28
4.6	Diagrama de classes . . . . .	31
4.7	Diagrama d'activitat . . . . .	36
4.8	Captura de pantalla . . . . .	37
4.9	Captura de pantalla . . . . .	38
5.1	Xarxes públiques - Número matrius rebudes per diferents dominis	42
5.2	Xarxes públiques - Temps càlcul de matriu per diferents dominis .	42
5.3	Xarxes privades - Número matriu rebudes per diferents dominis .	44
5.4	Xarxes privades amb bucles . . . . .	46
5.5	Xarxes privades amb bucles - Dades entorns amb 15 dominis . . .	47
5.6	Xarxes privades amb bucles - Dades entorns amb TTL 20 . . . . .	47



# Capítol 1

## Introducció

La unió de la informàtica i les telecomunicacions ens va aportar una nova manera d'afrontar la compartició de dades i la transmissió d'aquestes en general. La “informàtica” es defineix com el conjunt de ciències, tècniques o activitats relacionades amb el tractament automatitzat de dades, i la “telecomunicació” com qualsevol transmissió, emissió o recepció de signes, senyals, escrits, imatges, sons o informacions de qualsevol natura, per cable (elèctric o òptic), radioelectricitat, mitjans òptics o d'altres sistemes electromagnètics. L'associació d'aquests dos conceptes ens va atorgar la ciència de les xarxes de computadors, que ens permeten tindre sistemes d'informació i sistemes informàtics cada cop més eficients i competitius.

Des de fa uns quants anys, el grup de recerca del SeNDA ha estat investigant sobre la seguretat en xarxes de computadors en sistemes informàtics envers el tractament de situacions d'emergència. Dins d'aquest marc d'investigació el grup de recerca SeNDA ha estat treballant amb diferents projectes com pot ser el MedIGS [5], o el SCRISSE [7], a més va fer una distinció [6] de les diferents zones que existeixen dins un escenari d'emergència. La classificació es distingeix en 4 zones. La distinció comença per la zona 0 que es representa com l'epicentre de la catàstrofe i engloba tota l'àrea afectada per aquesta, sent la zona susceptible a albergar víctimes. Tindrem després la zona 1, que és la zona on es situa

la infraestructura d'emergència per donar tractament als afectats de les diverses zones 0. La zona 2 és la zona totalment externa a la catàstrofe ja que posseeix la infraestructura fixa necessària per donar suport complet a les víctimes. Per últim tindrem la zona 3 és on es contindrà tot el conjunt de la infraestructura capaç d'aportar informació sobre les víctimes, prèvia a la catàstrofe, als centres de la zona 2.

Donada la classificació i les aplicacions comentades, destinades a tractar aquest tipus de situacions d'emergència, es va veure que seria interessant tractar la interrelació entre els diferents elements que participen dins d'un entorn sobre la zona 3. Podem apreciar que dins l'última zona existeixen certs elements que formaran la infraestructura. Aquests elements seran organitzacions com hospitals, parcs de bombers o comissaries de policia. Organitzacions que donaran suport a l'escenari. La idea consisteix en fer que les entitats es puguin interrelacionar entre elles. Aquesta via d'investigació ha estat tractada envers els mecanismes de interoperabilitat i control d'accés aplicats a situacions d'emergència, considerades com escenaris multi-domini, degut les diferents organitzacions que hi participen.

La idea principal és continuar el treball realitzat, estudiant la via d'investigació tractada i implementant tots els mecanismes de interoperabilitat i control d'accés dins d'una eina de simulació. Volem que l'eina de simulació ens permeti integrar els elements bàsics que participen dins d'una situació d'emergència, per després simular el contingut de l'escenari. Tot això per tal de poder valorar d'una manera més real els mecanismes investigats.

El nostre projecte agafa com a punt d'inici l'estudi i el treball realitzat per part del SeNDA sobre el tema de les situacions d'emergència. Com a base sòlida per començar a treballar disposem de la tesina presentada per Carlos Martínez García [9]. En aquest document es realitza un estudi focalitzant la problemàtica envers dues àrees, el control d'accés per entorns multi-domini i la flexibilitat i interoperabilitat en el control d'accés. El document mostra que treballar amb aquest tipus d'entorn multi-domini pot reflectir d'una manera realista el tractament d'un

escenari amb una situació d'emergència, on hi hauran un cert nombre d'elements diferents.

L'estudi també proposa detalladament els termes d'un sistema capaç d'aportar control d'accés a escenaris multi-domini amb la flexibilitat i interoperabilitat necessària per situacions d'emergència. El sistema dissenyat reflecteix l'essència dels entorns amb diferents entitats, fent viable la interrelació d'organitzacions dins l'escena. Donada l'experiència del grup SeNDA envers el tema, el projecte part d'una descripció detallada dels requeriments funcionals, on s'exposen quines són les seves necessitats, incloent els elements i mecanismes que són necessaris per a fer viable el desenvolupament. Així es continua el treball d'aquesta branca d'investigació, fent realitat el sistema dissenyat en el document, desenvolupant una eina de simulació en entorns multi-domini, incloent els mecanismes de interoperabilitat i control d'accés.

El motiu que ens porta a realitzar aquest projecte es basa en què, donada l'evolució de la investigació ens veiem amb la necessitat de tindre un entorn de simulació per l'estudi de polítiques de interoperabilitat. Ens trobem així donat que aquesta és una línia d'investigació que es troba en un fase poc madura i per tant és necessari realitzar algun tipus d'implementació per comprovar la seva viabilitat. Això no es pot aplicar encara en escenaris reals, donat la dificultat que suposaria fer-ho, en canvi, disposar d'una eina de simulació ens permetria començar a valorar moltes coses sobre l'investigació, a més de donar-nos informació que podríem interpretar i valorar per recerques futures.

Donades totes aquestes premisses inicials i analitzades les necessitats exposades al document de requeriments, podem exposar de manera detallada quins seran els objectius inicials del projecte.

- **Realitzar un estudi detallat sobre el treball previ per tal d'aplicar aquests conceptes a la nostra aplicació.** L'objectiu és tindre ben clar quines han estat les vies de desenvolupament del document previ per tal d'analitzar tot

el contingut envers el nostre projecte.

- **Realitzar un disseny detallat dels elements que formaran part del projecte.** L'objectiu es basa en avaluar les necessitats, limitacions, associacions i dependències dels diferents mòduls per tal d'elaborar un disseny sòlid de l'estructura del nostre projecte.
- **Realitzar una implementació del treball, obtenint així l'eina de simulació en entorn multi-domini;** Sempre basant-nos en els dissenys realitzats. Tractarem també d'avaluar al final tot un conjunt de proves per tal de comprovar si l'aplicació dóna els resultats esperats i si aquest respon a les expectatives.
- **Elaborar i executar un banc de proves per validar l'aplicació i extreure conclusions dels resultats.**

Un cop hem vist quins han estat tots els objectius inicials, analitzarem com es compona l'estructura d'aquest document.

- Capítol 2 : Es realitzarà en aquest punt, una explicació sobre els conceptes bàsics per tal de que el lector pugui assolir tota la informació necessària que necessitarà per a comprendre els següents capítols.
- Capítol 3 : S'aportarà un anàlisi detallat i una valoració de tots els elements que seran necessaris per tal de poder portar aquest treball a bon terme. S'avaluarà a partir d'aquest anàlisi la viabilitat del projecte.
- Capítol 4 : Ens trobarem en aquest capítol tota la descripció sobre els diferents mòduls implementats dins el projecte. La descripció es realitza sota els punts de disseny i implementació. L'objectiu és atorgar al lector la possibilitat de veure com s'han realitzat els diferents mòduls i com s'ha plasmat tot l'estudi previ sobre una aplicació real.
- Capítol 5 : S'elaboraran un conjunt de proves i simulacions que s'introduiran i s'executaran sobre la nostre aplicació. A partir d'aquí es recolliran totes les dades per a una posterior avaluació.

- Capítol 6 : Realitzarem en aquest capítol les conclusions del treball realitzat i farem una valoració del conjunt de resultats obtinguts. Conclourem si el treball respon a les expectatives de l'estudi previ.





# Capítol 2

## Fonaments

Definirem a continuació tots els punts rellevants per tal de facilitar la correcta entesa d'aquest document. S'exposaran, en una primera secció, els conceptes sobre els que treballa i es basa el projecte que són el control d'accés i la interoperabilitat. A la segona secció s'exposaran els elements sobre els quals treballa el nostre simulador. És aquí on es realitzara una explicació per tal de poder entendre la funcionalitat d'aquests elements dins l'estructura del programa. Per últim es mostraran quines són les tecnologies que s'han utilitzat per tal de poder realitzar el simulador.

### 2.1 Conceptes

Cal afegir, respecte el paràgraf introductori del capítol, que tots els conceptes que s'expliquen a continuació estan extrets del document del SeNDA [9]. A l'estudi s'exposa detalladament quina és la descripció d'un sistema capaç d'aportar control d'accés a escenaris multi-domini amb la flexibilitat i interoperabilitat necessària per situacions d'emergència, dins la descripció apareixen tot un conjunt de conceptes que seran els que s'exposaran a continuació.

### 2.1.1 Control d'accés

Comencem aquí amb el concepte del control d'accés, un element molt important dins el nostre simulador. Per definició, el control d'accés és tot aquell sistema que té l'habilitat de permetre o denegar l'accés a un recurs específic per part d'una entitat determinada. On les entitats són aquells elements capaços d'executar accions sobre els recursos de dins del entorn. Les entitats normalment són els subjectes o usuaris. Els recursos són aquells elements passius que reben les accions per part de les entitats. Per fer-nos la idea, podem representar un recurs com una base de dades o una cua d'impressió.

Podem apreciar que el control d'accés és com un control d'autorització. Fixem-nos una mica més en detall en què consisteix aquest concepte. El control d'accés ens ha d'aportar unes premisses sobre les dades:

- **Secret** : Fa referència a l'objectiu de mantenir la privacitat sobre un recurs.
- **Integritat** : Fa referència a l'objectiu de tindre uns recursos fiables, no corromputs, lliures de modificacions malintencionades.

Aquests dos termes ens diuen que, dins un entorn on volem aplicar control d'accés, voldrem que els recursos mantinguin la seva integritat i que no puguin ser accedits per qualsevol. En altres paraules, volem que l'accés a les dades estigui perfectament controlat, per tal de què un d'aquests accessos no permeti que el contingut sigui de domini públic o que aquestes dades siguin modificades o eliminades sense autorització. Si ens fixem una mica, veurem que tots dos termes estan completament relacionats, perquè per tal de poder obtenir una bona integritat de les dades s'ha d'assolir un alt grau de privacitat sobre aquestes, a fi de controlar qui pot modificar el contingut.

Basant-nos en el nostre entorn de treball, tindrem les diferents entitats que estaran representades dins el nostre simulador. Aquestes es componen per usuaris, perfils i recursos. Dins de cada una de les entitats, els perfils defineixen els permisos dels usuaris i s'utilitzen per especificar les polítiques de requisits que ha

de tenir un usuari per accedir a un recurs. Els usuaris de les organitzacions seran els què, en moments determinats, executaran accions sobre els recursos. Recursos dels quals són propietaris les pròpies entitats. Per aquest motiu ens interessa tindre un control d'accés que determini si l'usuari d'una entitat pot accedir als recursos d'una altra; Serà mitjançant la relació entre els perfils de les diferents organitzacions, com es podrà establir quina similitud de perfil té cada usuari respecte a la resta d'organitzacions. Més endavant, a la secció de “Disseny i implementació”, s'exposara com es realitza aquesta conversió d'atributs entre entitats i quines són les polítiques implementades.

### **2.1.2 Interoperabilitat**

Tractarem ara la interoperabilitat, un concepte molt estès avui dia, com es pot apreciar per exemple amb les xarxes socials o el comerç electrònic [10]. Concepte el qual tracta la relació i el treball comú entre les diferents entitats que conviuen dins un mateix escenari. Aquest és un concepte que penja del camp de les xarxes de computadors o de comunicacions.

Implementar la interoperabilitat dins el nostre projecte, ens dona l'oportunitat d'interconnectar totes les organitzacions disponibles del nostre escenari. La idea de què treballin totes les entitats juntes i interrelacionades, atorga funcionalitats i noves oportunitats dins del camp, fent que compartint la seva informació individual envers la col·lectiva estiguem al davant d'un entorn multi-domini eficient per al ben comú. Sota el punt de vista d'aquest concepte, aplicat al nostre simulador, podem dir que cada una de les organitzacions presenta una estructura d'organització interna reflectida en la jerarquia de perfils/rols dels usuaris associats a l'entitat. Generalment, aquestes estructures són independents entre les diferents entitats. És per això que la interoperabilitat tracta de generar un sistema comú a partir les organitzacions definides a l'escenari, per tal de generar una estructura global basada en aquests atributs. D'aquesta manera es facilitarà la comunicació entre les diferents entitats.

Comentarem, respecte la interoperabilitat, dos principis importants que han de mantindre totes les entitats dins del nostre entorn. Són uns punts que afecten a la gestió de les polítiques d'accés de cada sistema per separat.

- **Principi d'autonomia :** Dicta que cada sistema ha de ser administrat per separat, decidint quins privilegis atorguen a cada usuari. Aquests privilegis no es poden veure disminuïts pel pacte de col·laboració entre entitats.
- **Principi de seguretat :** Dicta que un usuari no pot guanyar privilegis dins del seu propi sistema través de un pacte de col·laboració entre entitats.

Amb la finalitat de relacionar aquest conceptes amb el control d'accés comentarem que, dins de cada entitat es definiran unes polítiques d'accés a les dades, per a que es pugui determinar si és viable que un usuari d'una altra entitat pugui o no accedir a aquestes. Es disposa de la definició d'atributs dins de cada entitat, per poder definir els rols o permisos associats als subjectes que pertanyen a la entitat en qüestió. Sota aquestes premisses, veiem que seria interessant poder definir graus dins d'aquesta autorització. Per fer això viable, dins del control d'accés, es treballa amb la conversió d'atributs. La idea és realitzar una conversió equivalent dels atributs de una entitat origen, cap a una destí. La conversió vindria donada per un valor, anomenat factor de conversió, que té un rang entre 0 i 1, on 0 indicaria una autorització mínima i 1 un valor màxim. És una bona forma de donar independència a les entitats i fer que només estiguin pendents de la conversió d'atributs.

Caldria puntualitzar el terme pacte de col·laboració, que ha aparegut quan es comentaven els punts que afecten a la gestió de les polítiques d'accés. Aquest pacte de col·laboració estableix una relació directa entre entitats i és en aquest moment quan es realitza una conversió equivalent dels atributs entre dos entitats. En aquest punt s'estableixen els factors de conversió entre els seus respectius atributs. És necessari afegir que, si es realitza una conversió d'atributs no s'obliga a les organitzacions a canviar el seu sistema, ja que els usuaris que venen de fora, al

obtenir atributs locals es poden sotmetre a polítiques de control d'accés de l'organització.

Encara que existeixen altres vies en desenvolupament [8], la interoperabilitat a nivell d'atribut és considerada la via que millor s'adapta als escenaris reals. En comptes de tractar la interoperabilitat amb un conjunt d'atributs universals compartits per totes les entitats, tenim que cada organització disposa dels seus atributs on treballa de forma autònoma, on cada atribut disposa d'una informació depenent del seu context disposant d'aquesta manera d'un entorn més real.

## 2.2 Elements del l'escenari

Detallarem en aquest punt els elements sobre els que treballem amb el nostre simulador, fent una vinculació natural d'aquests elements respecte els punts esmenats al capítol d'introducció d'aquest document.

### 2.2.1 Dominis

Des de l'inici del document hem parlat de diferents conceptes, però sempre des d'un punt d'alt nivell dins del significat del concepte o sempre destacant la seva vessant sobre un punt de vista enfocat al món real. Un d'aquests elements és, per exemple, el d'organitzacions. Com s'ha comentat anteriorment, aquests elements serien les diferents institucions dins el nostre entorn.

Respecte un punt de vista més proper al nostre simulador, parlarem de l'element "Domini", per a referenciar a les organitzacions que interaccionen dins el nostre programa. A l'igual que les organitzacions tenien les seves característiques, com una estructura interna definida per els perfils que fixaven els permissos del usuaris, els dominis també seran representats amb els mateixos trets característics envers la seva estructura. Cal tenir clar que, cada organització dins el món real, tindrà una relació única i directa sobre un element domini. L'element domini serà

important pel fet de què cada objecte instanciat tindrà associat un conjunt d'atributs que reflecteixen la seva estructura, també disposa d'un conjunt d'usuaris que pertanyen a l'entitat en qüestió. Els atributs ens definiran els rols o els perfils dels usuaris vinculats al domini. Podem valorar que l'element domini jugarà un paper molt important dins el nostre simulador.

### **2.2.2 Funcions i Escenaris de conversió**

Ja s'ha introduït abans el concepte de conversió d'atributs i com ja sabem, tot això, ve donat per les premisses de interoperabilitat i control d'accés que volem implementar al nostre simulador. Serà necessari, abans de tot, aclarir la nomenclatura a utilitzar. Per anomenar l'element que ens donarà el valor de la conversió podrem utilitzar les denominacions matriu de conversió, funció de conversió o política de conversió. Per definir les metodologies a utilitzar per obtenir aquestes conversions entre els diferents dominis utilitzarem la denominació escenari de conversió.

#### **Dominis dins de l'entorn**

Per poder aplicar l'interacció entre les diferents organitzacions necessitem, en un primer moment, tenir definits sobre el nostre entorn un conjunt de dominis i que aquests tinguin associats un conjunt de relacions usuari-atribut. Un cop tenim això, voldrem associar els diferents dominis per tal de que els seus usuaris puguin accedir a les dades d'un altre domini diferent. Tindrem llavors les organitzacions dins el nostre escenari definides per un conjunt d'atributs. Tindrem també altres entitats / dominis, que tindran definits uns altres atributs que poden ser iguals o diferents. És aquí on es necessita el concepte de la conversió d'atributs. Això ens ve pel fet de voler fer que un usuari del domini X accedeixi a recursos del domini Y. Podríem fer una relació directa entre els dominis per tal de poder tramitar la informació. És en aquest moment quan volem fer la interacció on apareix el problema.

Atributs	Metge	ATS	Infermer
Metge	0.8	0.6	0.3
Zelador	0.1	0.4	0.8
Infermer	0.2	0.5	1

Taula 2.1: Matriu de conversió

### Funció de conversió

Dins d'una entitat els usuaris tenen uns privilegis i/o permisos per accedir a les seves dades que, en última instància, són atorgats mitjançant la concessió d'atributs. És per això que necessitem saber, per el conjunt d'atributs que té un domini, quina correspondència tenen aquests atributs a l'altre domini. Això ens serà molt útil per trobar la compatibilitat de permisos entre usuaris de diferents entitats per tal de poder establir a quina informació poden accedir / modificar els usuaris. Aquesta conversió s'anomena "Funció de Conversió" i per poder realitzar-la utilitzarem el que anomenarem com els "Escenaris de Conversió". Les "Polítiques de Conversió" establiran les matrius, (o funcions de conversió), que relacionen els atributs d'un domini amb els atributs de l'altre domini. Per exemple, es mostra a la taula 2.1 una matriu on disposem els atributs de dos hospitals. El primer hospital té associats els següents atributs: Metge, ATS i Infermer. El segon hospital té definits els atributs: Metge resident, Zelador i Infermer. Llavors amb aquests atributs i les polítiques de conversió es podria obtenir una matriu com la que es mostra a la taula 2.1.

Aquesta matriu també pot ser vista com una "fuzzy relation" entre els conjunt d'atributs de les dues organitzacions. El contingut d'aquest element mostra uns números que indiquen la similitud / correspondència entre els atributs. S'utilitza una matriu per a la representació donat que és la única manera de fer composicions d'aquest tipus d'una manera tan fàcil i senzilla. A més disposa d'una bona versatilitat envers els càlculs i la seva estructuració permet una fàcil assimilació visual dels continguts.

**Entorns de conversió en grups de dominis**

Aquest exemple que s'acaba d'il·lustrar reflecteix la relació feta entre un domini X amb un domini Y, on estem relacionant els seus atributs entre ells. Aquí hem resolt el problema de la conversió, ja tenim una manera d'associar els dominis, però que succeïx si arriben més dominis al nostre entorn?, per exemple Z i T, i si, a més a més, volem que des de qualsevol domini es pugui accedir a qualsevol altre dins l'entorn?. Idealment qualsevol domini pot tindre una funció de conversió per la resta de dominis, però significa que hauríem de definir en total  $n^2 - n$  funcions de conversió diferents. Això fa que no sigui gens escalable. Aquest problema es resol mitjançant l'encadenament de funcions de conversió mitjançant els nostres escenaris de conversió, és per això, que idealment tindrem funcions de conversió entre dominis, inicialment declarades i formalitzades per part dels dos dominis que formen un vincle directe. Serà a partir d'aquests vincles directes on nosaltres podrem obtenir noves funcions de conversió saltant de domini en domini a través d'aquestes funcions de conversió ja existents, mitjançant els escenaris de conversió, sense la necessitat de crear noves relacions entre dominis per totes les entitats existents. És a dir, l'idea bàsica és obtenir la funció de conversió entre X i Z, donada una conversió entre X i Y i una altra entre Y i Z. Els escenaris de conversió que permetran fer això viable són:

- **Xarxes públiques de conversió :** Dins d'aquest mètode cada domini coneix totes les funcions de conversió d'atributs que hi ha a l'escenari. Cada domini ha de ser capaç de realitzar qualsevol procés de conversió d'atributs.
- **Xarxes privades de conversió :** Aquí cada domini coneix només les funcions de conversió d'atributs que l'afecten directament a ell. El procés de conversió d'atributs requereix un pas de missatges entre els diferents dominis.



## 2.3 Tecnologia a utilitzar

Finalitzarem aquest capítol de fonaments exposant quines seran les eines ha utilitzar per tal de dur a bon terme la creació del simulador.

### 2.3.1 Java i Swing

Durant la definició del problema i de les estratègies a utilitzar durant el desenvolupament del projecte, no es va dubtar en quin llenguatge de programació s'utilitzaria. El llenguatge de programació Java és un llenguatge orientat a objectes de principi dels anys 90. És un llenguatge molt estès i molt utilitzat durant la carrera, els coneixements i l'experiència prèvia sobre aquesta eina eren suficients com per desenvolupar el projecte. Java ens reporta avantatges que ens evitarà molts errors durant el desenvolupament:

- Modularitat més simple que amb C o C++ donat el seu model d'objectes.
- No utilitza directament eines de baix nivell, com pot ser la manipulació de punters.
- Java disposa de una independència multi-plataforma, això significa que programes escrits en Java poden ser executats sobre qualsevol hardware. El codi en Java es genera com un "bytecode" i això s'executa sobre la Java Virtual Machine, que haurà de tindre instal·lada la mequina per a poder executar el codi.
- Java inclou una biblioteca gràfica anomenada SWING, molt estesa i molt completa per tal de poder realitzar l'interfície gràfica del nostre simulador. L'unió d'aquestes dues tecnologies, simples, accessibles i eficients va fer que escollir l'eina amb la que es treballaria no fos un problema.

### 2.3.2 XML

Parlem ara de la tecnologia XML. Les seves sigles signifiquen Extensible Markup Language. El XML és una metallenguatge extensible d'etiquetes i té com a objec-

tiu principal definir un estàndard per l'intercanvi d'informació estructurada entre diferents plataformes de una manera fàcil i ràpida. El XML és una tecnologia que ens permet emmagatzemar l'informació dins d'un fitxer d'una manera molt senzilla i ben estructurada. El contingut del fitxer esta dissenyat amb una estructura jeràrquica, per tal de fer l'organització molt fàcil a efectes de poder ser visualitzada o manipulada per una plataforma, ja sigui per desar l'informació o per obtenir-la.

Dins del nostre simulador utilitzarem la tecnologia XML per a poder emmagatzemar l'estat del nostre simulador en un moment determinat. És a dir, si disposem d'un entorn que estem simulant, on apareixen unes entitats amb una relacions fetes, podrem guardar tota la informació dins d'un fitxer XML per no perdre aquesta estructuració i per tal de poder recuperar-la. La tecnologia XML ens aportarà un benefici molt gran, donat que per a grans simulacions amb grans quantitats d'informació, podrem importar i/o exportar l'informació sense cap tipus de problema. Tot això implica que ens podrem estalviar molt de temps i fer que el simulador sigui més eficient.

# Capítol 3

## Anàlisi

Comencem ara un apartat molt important dins el desenvolupament del projecte. Tal com es va establir al inici s'elaborarà el treball en diferents apartats. Un cop s'han definit clarament els objectius i es tenen clars els conceptes relacionats amb el projecte, es començarà amb la part d'anàlisi. Aquest apartat està directament lligat amb la part “anàlisi de viabilitat”, on s'estudiarà i es desmembrarà el factor de que el treball sigui realitzable. Dins d'aquest punt serà interessant analitzar els diferents aspectes mesurables dins la viabilitat del projecte. Es valoraran d'una forma concreta i àmplia els diferents àmbits que ens acompanyaran durant el desenvolupament del treball, donant-nos una idea de quines seran els avantatges i els inconvenients que tindrem. El present estudi ens ajudarà a valorar en quin grau la realització d'aquest projecte és viable.

### 3.1 Anàlisi de viabilitat

Dins de l'apartat “anàlisi de viabilitat” es desglossarà el problema en quatre punts importants. La viabilitat d'un projecte no és una cosa que es pugui tractar directament sobre un únic punt. És interessant valorar els factors importants que afecten directament i indirectament sobre el desenvolupament del projecte. És per això que es filtrarà l'estudi per els punts referents a la viabilitat tècnica, econòmica, legal i temporal.

### 3.1.1 Viabilitat Tècnica

Sota les premisses que comporta una viabilitat tècnica dins el projecte, hem de valorar certs aspectes. En un primer moment valorem les eines que necessitem per tal de poder portar el treball a bon terme. Eines de desenvolupament que, en el nostre cas, seran el llenguatge de programació Java i la seva biblioteca gràfica Swing. Per tal de poder desenvolupar en aquest llenguatge necessitarem el compilador del llenguatge de programació i un IDE per a la edició i realització del codi. Per a poder fer ús d'aquestes eines necessitem un ordinador personal, PC, equipat amb un processador compatible amb repertori de instruccions x86. L'ordinador haurà de tindre un sistema operatiu GNU/Linux instal·lat, una Debian Lenny 5.0 o una Ubuntu 9.04 és perfectament vàlid. També seria bo disposar d'un sistema operatiu Windows XP o Windows Vista, per tal de fer possible l'anàlisi de l'aplicació sobre diferents plataformes. No serà necessari cap hardware extra.

Després de valorar tots aquests punts podem comentar que, pel que respecta a la viabilitat tècnica, la realització és completament viable. Tots els elements esmentats es poden aconseguir sense problemes i es tenen ja disponibles a l'abast de la mà. Això és així donat que les eines poden ser obtingudes a través d'Internet. Respecte els ordinadors i el seu sistema operatiu no comportarà problema donat que el departament disposa d'una sala on es disposen diferents ordinadors equipats per tal de poder realitzar tasques de desenvolupament, així com un despatx on es podran realitzar les diferents reunions personals. El departament disposa també d'una sala de reunions, on es poden portar a terme diferents sessions de control entre tots els projectistes i els seus respectius coordinadors.

### 3.1.2 Viabilitat Econòmica

L'anàlisi sobre els punts econòmics d'aquest treball seran realment senzills. Les eines de programació que s'utilitzaran, IDE i compilador de Java, són completament gratuïtes i no comporten cap cost per obtenció, ús o manteniment. Disposar d'un ordinador, com l'especificat a l'apartat anterior, que tingui un sistema opera-

tiu GNU/Linux instal·lat no serà difícil i no suposarà un cost addicional donat que com hem comentat, el departament disposa d'un laboratori amb PC's preparats. Obtindre un ordinador no suposarà un cost addicional. El departament disposa de la infraestructura necessària i no caldrà destinar-hi una inversió per cap element mencionat al apartat de viabilitat tècnica. El sistema operatiu GNU/Linux és gratuït a més, si s'ha de comprovar la operabilitat interplataforma de l'aplicació, el meu ordinador personal ja disposa de una plataforma Windows per a poder fer les proves.

### 3.1.3 Viabilitat Legal

La viabilitat legal sobre aquest projecte tampoc suposa una problemàtica molt gran donat que el software a utilitzar esta sota llicència GPL [11][12]. Això ens permet utilitzar el software sense problemes, a més implica que el nostre software també es podrà utilitzar sense problemes en el futur. També s'ha de tenir en compte, tal com explica Carles Martínez Garcia a la seva tesina de Màster [9], que en cas de tractar amb dades mèdiques reals, s'haurà de tindre en compte la Llei Orgànica de Protecció de Dades [13] a l'hora d'accedir a les dades mèdiques.

### 3.1.4 Viabilitat Temporal

Per últim ens trobem amb la viabilitat temporal. És aquí on s'analitza si els objectius i les tasques establertes en el moment inicial de la definició del projecte podem ser portades a terme. Es fa aquest anàlisi inicial sempre basant-se en termes de viabilitat temporal. És molt important diferenciar el que implica aquest punt, on es valora l'anàlisi temporal en els moments inicials del desenvolupament del treball, entre el punt que més endavant veurem, on es tracta de valorar de forma analítica si aquesta planificació temporal ha estat correcta.

Fases, tasques i temps. Aquests són els trets bàsics en que es basarà en nostre anàlisi. S'han definir en primera instància les tasques a realitzar associades a les fases principals, tot això sota el marcatge entre límits temporals. Podem apreciar

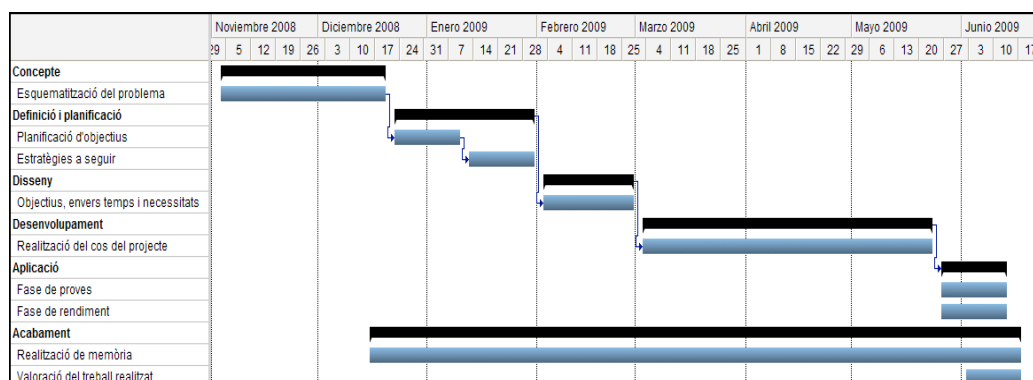


Figura 3.1: Diagrama de Gantt

a la figura 3.1 una taula on es mostra la planificació i ens indica si és viable en termes de temps. L'estructuració temporal que mostra la figura fa que el projecte, en termes de viabilitat temporal, sigui completament viable

# Capítol 4

## Disseny i implementació

Ens endinsem ara en un apartat on tractarem i analitzarem els elements clau del nostre projecte. Serà interessant per aquest punt, agafar els mòduls essencials per tal de desglossar i analitzar les seves parts, fent una ullada minuciosa a la seva estructura. En un primer moment ho farem des del punt de vista del disseny i després envers la seva implementació. En primer terme, ens centrarem en els mòduls que donen lloc a l'estructura bàsica del treball. Estructura que estarà per sota del nostre framework de simulació. Aquesta ens permetrà contenir tota l'estructura d'informació que ens farà falta per tal de donar suport, més endavant, a la resta de mòduls del projecte. Analitzarem en segon terme, els mòduls que ens aporten la funcionalitat interactiva i d'interès sobre el nostre framework, detallant el seu funcionament i tot el que ens pot aportar.

### 4.1 Disseny mòduls principals

Dins la present secció agafarem els mòduls principals desenvolupats i els analitzarem. Els observarem detingudament per veure quines són les seves característiques, les seves particularitats, quines són les seves limitacions que ens poden aportar i quina és la interconnexió entre aquests. Donem una petita definició dels elements:

- **Escenari** : És el que representa l'entorn on tindrem tots els dominis.

- **Dominis** : És el que representa a les entitats i organitzacions que tindrem dins l'escenari.
- **Atributs** : Són els valors que defineixen els perfils/rols dels usuaris dins dels dominis.
- **Funcions de conversió** : Són les matrius que ens donaran l'equivalència d'atributs entre dominis.

#### 4.1.1 Mòdul: Escenari

Donades les circumstàncies, premisses i necessitats definides respecte les necessitats, ens trobem ara en una situació on es requereix d'un element, dins la nostra estructura, que ens permeti cimentar el conjunt d'objectes del nostre framework. És per això que ens veiem amb l'obligació de crear l'escenari, un element absolutament necessari que ens dóna l'oportunitat definir tots els objectes necessaris per a poder treballar. Els elements que tractarà directament l'escenari són els dominis, per això es disposarà d'un vector on es contindran tots els dominis creats per l'usuari on, amb aquest vector d'informació, l'escenari enviarà la informació al mòdul de dibuix i als mòduls dels nostres escenaris de conversió. L'escenari ha estat dissenyat amb la intenció de tindre una bona base de l'arquitectura donat que, des de l'escenari s'organitzen la resta de mòduls i si no hi ha escenari no hi ha simulador.

#### Limitacions i restriccions

L'accés al contingut dels dominis per part de l'escenari es troba limitat. Cal tenir en compte que el mòdul escenari disposa del vector d'entitats per al seu ús, però s'ha implementat de tal manera que l'escenari només pugui modificar el contingut del vector amb la creació i/o eliminació dels dominis. Els dominis són creats o eliminats sota la direcció del usuari. No hi haurà cap limitació respecte al nombre d'objectes domini que pot contenir el vector.



### 4.1.2 Mòdul: Dominis

En segon terme parlem del mòdul de dominis. Aquesta peça és la base de la nostra estructura d'informació. Aquí és on podrem definir els nostres nodes dins un vector per el nostre escenari. Totes les organitzacions que vulguem incloure dins el nostre simulador tindran el seu node característic dins d'aquest vector. A l'hora de la definició es podran constituir tots els dominis desitjats, per ser definits es demanaran un conjunt de trets característics per tal de poder configurar el node i poder identificar-lo con l'usuari estableixi. S'introduirà el nombre, un valor identificatiu i un conjunt d'atributs que especificarem al següent punt. Com ja sabem els atributs són els perfils i rols que identifiquen els usuaris associats a l'entitat i podrem catalogar els permisos envers aquests perfils. El domini serà la relació única i individual d'una organització real envers el nostre simulador. El domini, per definició, té associat un vector d'atributs que el caracteritzen i un vector de funcions de conversió.

#### Limitacions i restriccions

El domini podrà accedir a totes les seves dades però amb limitacions. Un domini podrà afegir elements al seu vector d'atributs en el moment que s'està creant el propi domini, un cop establert el domini a l'escenari aquest no podrà afegir o eliminar cap atribut, però si que podrà consultar-los. Expliquem el per què al següent punt. Respecte les matrius de conversió la cosa és diferent donat que el mòdul té mètodes per accedir a les matrius i modificar el contingut. Això és degut a que l'objecte que té més moviment dins el nostre simulador és la matriu de conversió. El domini té la possibilitat de crear o eliminar dades, dins del marc comentat, però no ho podrà fer amb el contingut d'altres dominis que no sigui ell mateix. Remarquem això donat que és aquí on es planta la base de les premisses de control d'accés i de interoperabilitat. És important remarcar que estem tractant amb el mòdul que conté tota la informació relativa a les dades que volem simular.

### 4.1.3 Mòdul: Atributs

Parlem ara del mòdul referent als atributs. Possiblement és el mòdul més senzill de tots els quatre que estem exposant aquí. L'objectiu del mòdul "Atributs" és el fet de perfilar o determinar les característiques del domini, es defineix d'aquesta manera l'estructura interna del domini reflectida pels atributs declarats. A l'hora de crear un, l'usuari vincularà al domini tot un conjunt d'atributs que definiran els rols característics que pot contenir o es poden associar al node. La creació d'atributs no té límits durant el procés de creació del domini, donat que són molt importants per a la caracterització de l'organització i per la pertinent simulació. Sabem que els atributs defineixen els perfils i rols que pot contenir el domini en qüestió.

#### Limitacions i restriccions

La limitació sobre aquest mòdul ens vindrà per part de l'edició dels atributs del propi domini que estarà deshabilitada. L'associació dels atributs es realitza només en el moment de la creació del domini objectiu. Un cop finalitzada la creació no es poden modificar/afegir/eliminar els atributs d'un domini. L'única solució és eliminar el domini directament i fer-ho de nou. Això està implementat d'aquesta manera per fer-ho més real envers el món real. Sabem, segons s'ha exposat a la secció de interoperabilitat dins el capítol de fonaments, que a la realitat les entitats poden formalitzar funcions de conversió de forma directa d'entitat a entitat, pactant els graus de conversió entre els seus respectius atributs, és per això que si iniciem un escenari, on tenim uns dominis amb un conjunt de funcions de conversió prefixades, no seria just ni s'ajustaria a la realitat el fet de poder modificar el contingut dels atributs. Si una entitat necessita canviar els seus atributs llavors haurà de canviar les funcions de conversió i per fer això haurà de tornar a pactar-les amb les altres entitats. Sempre hem de tenir present que, en aquest punt, estem parlant de les matrius pactades entre entitats, no les obtingudes a partir de les metodologies que proporcionen els nostres escenaris de conversió.

#### 4.1.4 Mòduls: Funcions de conversió

Aquest és el mòdul que ens dona la interoperabilitat entre els diferents nodes. Contindrà la informació associada entre diferents dominis i es crearà una funció de conversió, (o matriu de conversió) només per cada dos dominis. La funció de conversió en sí conté un valor que reflexa la conversió d'atributs entre dos nodes, és a dir, si volem associar el domini A amb el domini B llavors es crea una matriu, on els atributs d'A seran les files i els atributs de B les columnes. Llavors per cada casella es marca l'equivalència d'atributs entre els diferents nodes, el valor d'equivalència és un valor amb un rang d'entre 0 i 1. Aquest valor indica que com més baix, menys equivalents són els atributs.

Com ja s'ha esmenat al capítol de fonaments, s'utilitza una matriu pel fet de que és un element que ens permet fer composicions fàcilment, a més permet fer operacions amb versatilitat i la seva estructura és ideal per a una fàcil interpretació visual. Respecte a la seva estructura interna hem de comentar que, com més atributs continguin més gran serà la matriu. També és important notar que, una funció de conversió entre el domini A i el domini B no significa que sigui bidireccional, és a dir, sota la premissa que treballem amb grafs dirigits, amb aquesta associació només hi ha camí de A cap B, però no a l'invers. Per fer-ho caldrà realitzar una associació de B cap a A.

El motiu pel qual aquesta associació no és bidireccional és perquè, donada la present situació, quan un domini A decideix col·laborar amb un altre B, el domini A assumeix un risc, donat que deixarà que altres usuaris utilitzin els seus recursos. Aquest ha d'acceptar el risc i assegurar-se de que és el domini A qui emmagatzema la matriu. Les matrius poden ser perfectament simètriques però cada entitat ha de guardar una al domini destí. Per tant, una única matriu no és suficient, calen dos matrius per obtenir la bidireccionalitat de la relació. Un altre motiu és que potser les entitats a associar utilitzen estructures internes diferents, és a dir, podríem tindre una relació amb entitats on una d'elles no disposa de recursos per oferir.

Atributs	B1	B2
A1	0.2	0.4
A2	0.6	0.8

Taula 4.1: Matriu de conversió entre A i B

Atributs	C1	C2
B1	0.4	0.6
B2	0.3	0.5

Taula 4.2: Matriu de conversió entre B i C

### Com es calculen els valors de conversió

Hem introduït molts conceptes, però no hem vist com es calculen els valors de conversió. Imaginem que tenim 3 dominis A, B i C. I els dominis A i B tenen una funció de conversió i B i C una altra. Llavors nosaltres decidim trobar la relació entre A i C. L'algorisme s'anomena Max-Min i s'utilitza per fer composicions de relacions dins la teoria de conjunt difusos. Exposarem primer l'algorisme de forma teòrica i després el procediment implementat a l'aplicació. L'algorisme teòric es basa en diferents punts. Farem un exemple per la conversió entre A i C. Exposem les dues matrius de conversió entre els dominis A i B i entre els dominis B i C.

Exposem llavors els passos. Primer desglossem l'escenari, figura 4.1. Després agafem el primer atribut d'A i cerquem els camins possibles fins el primer atribut de C. Per cada camí obtingut s'agafa el valor més petit que conté el camí en qüestió, figura 4.2. En l'últim pas s'agafa el valor més gran dels valors recollits dels diferents camins, figura 4.3. La figura 4.3 ens indica també quin és el valor de conversió entre el primer atribut d'A i el de C. Per obtenir tota la matriu de conversió caldrà fer aquest passos per tots els atributs del domini d'A respecte tots els atributs de C. Avaluem, per exemple, amb les figures 4.4 i 4.5, la conversió d'atribut del segon atribut d'A amb el segon de C.

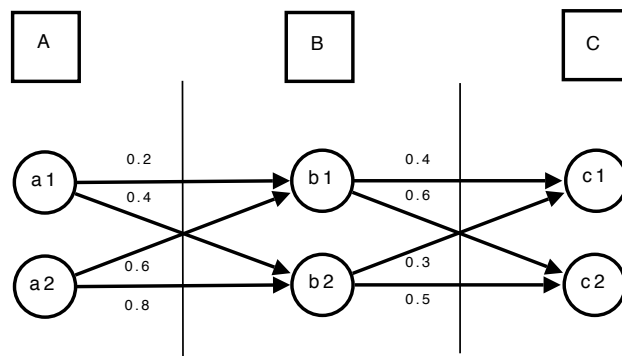


Figura 4.1: Conversió d'atributs - Escenari

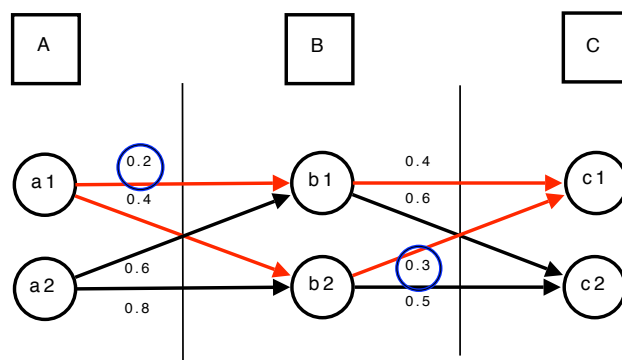


Figura 4.2: Conversió d'atributs - Selecció valors mínims en camins

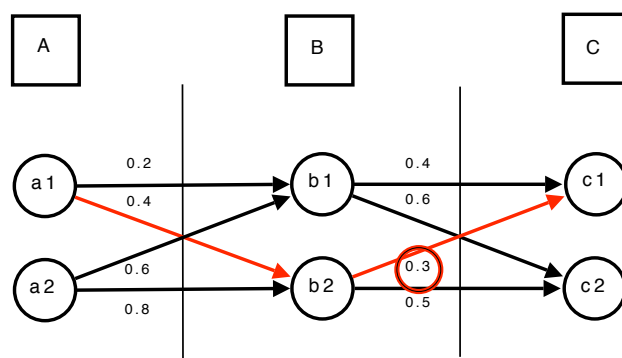


Figura 4.3: Conversió d'atributs - Selecció valor màxim dels camins

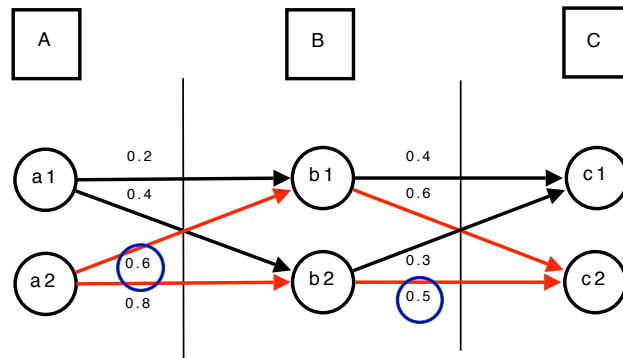


Figura 4.4: Conversió d'atributs - Selecció valors mínims en camins

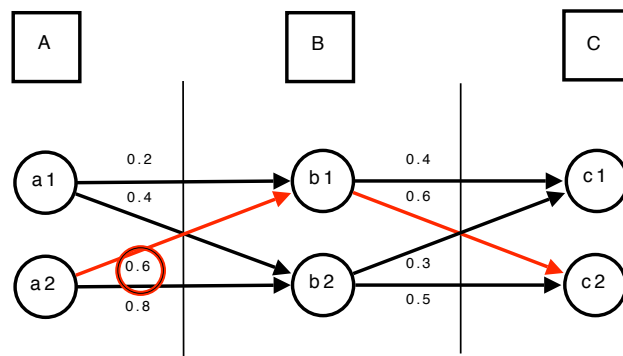


Figura 4.5: Conversió d'atributs - Selecció valor màxim dels camins

Ara exposem el procediment implementat a l'aplicació. Agafem cada fila de la primera matriu i la comparem amb cada columna de la segona matriu. Això és possible, donat que el número de columnes de la primera matriu és el mateix que de files de la segona matriu, donat que B ens fa de connector entre els dominis. Procediment:

- Agafem la primera columna i la comparem amb la primera fila.
- Agafem els primers elements de cada llista i ens quedem amb el més petit, després agafem els segons elements de la llista, els comparem i ens quedem amb el més petit, repetim la mateixa operació per la resta d'elements d'aquestes llistes.
- Un cop ho hem tramitat, dels valors mínims que hem obtingut ens quedem amb el més gran. Acabem d'obtenir el valor de conversió del primer atribut d'A amb el primer atribut de C.
- Fem això per la resta de files i de columnes, fent aquest procediment de comparació de cada fila de la primera matriu per totes les columnes de la segona.
- Al finalitzar tindrem la matriu de conversió entre A i C.

## 4.2 Disseny mòduls funcionals

Explicarem aquí, tot el que fa referència als mòduls que donen funcionalitat al simulador. Aquests elements són els que realitzen tots els càlculs, ho fan a partir de tota la estructura d'informació que hem pogut construir gràcies als mòduls principals, definits al punt anterior. Per aquest projecte, donades les especificacions inicials, s'han definit dos tipus de simulacions dins del marc del projecte.

Les xarxes públiques i les xarxes privades tramiten la mateixa informació donant el mateix resultat en la majoria de casos, el que canvia és la manera d'interpretar el mateix escenari. Aquesta diferenciació bé donada pel fet de que no és

necessari compartir les relacions entre dominis, (o relacions de conversió), com tampoc és necessari compartir la seva existència. Els mòduls poden realitzar la seva funció gracies a l'enviament de missatges entre els diferents dominis, cal dir que els missatges que s'envien els dominis entre ells són objectes del tipus *MatriuConversio*. L'objecte *MatriuConversio* ens dona una bona estructura per albergar tota la informació necessària per fer viable que aquests mòduls facin la seva funció.

La informació que conté l'objecte *MatriuConversio* ens serà molt útil, donat que aquest objecte a més de contenir l'estructura de matriu per guardar la funció de conversió, també inclou informació sobre el TTL, l'identificador del domini origen, del domini destí, del domini provisional. El domini provisional és l'identificador del domini per on acaba de passar. També tindrem una llista per referenciar els dominis recorreguts.

Abans d'entrar en aquest punt, mostrem la figura 4.6 on podem visualitzar els mòduls comentats dins el diagrama de classes en UML [1] del projecte. Dins la figura 4.6 podem apreciar els mòduls més importants dins la nostra aplicació, referenciem l'infraestructura d'informació amb les classes “Escenari”, “Domain”, “Atributs” i “*MatriuConversio*”, amb els seus atributs més destacables dins l'estructura d'informació. També es veuen refenciats la classe de l'entorn gràfic i les classes dels escenaris de conversió, les xarxes públiques i privades.

### 4.2.1 Mòdul: Xarxes públiques

Les xarxes públiques, o xarxes de conversió pública, són les xarxes on totes les relacions entre dominis i totes les funcions de conversió són conegudes per tots els nodes que formen l'escenari. Aquí tots els dominis poden formalitzar una possible conversió d'atributs amb qualsevol altre domini de dins del escenari, realitzant tot el procés determinista d'una forma eficient a expenses dels elements de privacitat



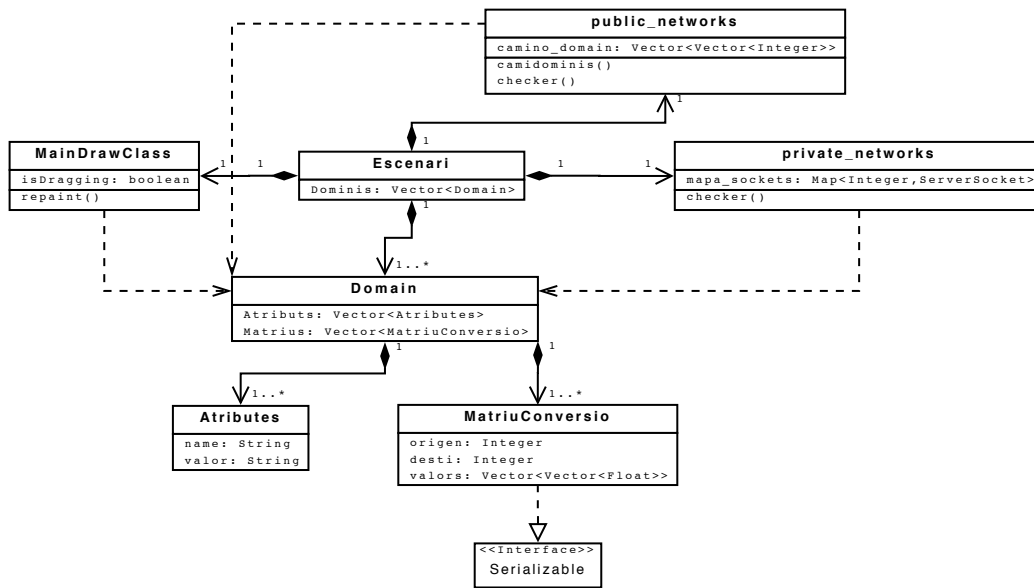


Figura 4.6: Diagrama de classes

dins l'entorn.

Per a poder realitzar una funció de conversió entre diferents dominis, per exemple entre A i B. El primer que hem de fer és buscar tots els camins possibles entre aquests dos dominis A i B. L'algorisme utilitzat comença a recórrer, per cerca en profunditat, des del destí fins a trobar l'origen, un cop arriba al origen comença a guardar el camí. Es va fer així per tal d'obtenir els camins anotats des de l'origen fins al destí. A l'hora de tornar enrere, si ha arribat a un node fulla, busca altres possibles camins al nou nivell. L'algorisme no torna directament al destí per completar el camí. Es van buscant tots els camins a l'hora. És interessant remarcar que l'algorisme descarta tots els camins on hi ha bucles, és a dir, mentre va buscant el camí a través del graf, si es troba en un node per on ha passat descarta el camí en construcció i passa a buscar un altre. Aquest pas és molt important donat que sense la exploració de camins no seria possible assegurar el nombre màxim de relacions entre els nodes.

És important remarcar perquè els camins amb bucles queden totalment des-

cartats. El motiu és ben senzill, no hem de permetre, per cap motiu, que un usuari utilitzi qualsevol mètode per a guanyar privilegis envers el seu perfil. No ha de ser viable, en aquest cas, que un usuari comenci a provar si, per altres camins entre els dominis, podria obtenir un grau de conversió més alt respecte al seu atribut. És per aquest motiu que es descarten els camins amb bucles per tal de que no hi hagi opció a que això succeeixi. Això es fa per tal de mantenir els principis de interoperabilitat i control d'accés a més que, augmentar els privilegis d'un usuari d'aquesta manera vulnera el principi de seguretat, principis els quals hem explicat al capítol de "Fonaments". Aquest punt dels camins i els bucles, s'implementa també per a les xarxes privades.

Continuem amb el comportament del nostre escenari de conversió. Un cop tenim una llista amb tots els camins possibles entre els dominis A i B per cada camí obtingut tindrem una funció de conversió, és a dir, per cada camí llencem des del origen una ordre de càlcul de la funció de conversió corresponent al camí recorregut. Cada funció de conversió variarà depenent dels diferents dominis que formen part del camí en qüestió. El procediment consisteix en agafar la matriu de conversió de cada domini que apareix al camí obtingut. Aquesta matriu de conversió es va calculant progressivament per cada domini de la llista, on finalment arribarà al domini destí. Un cop hem obtingut la matriu de conversió final la guardem per esperar a que acabi el procés sencer per a tots els camins.

Quan ja hem tractat tots els camins es computen els resultat per maximitzar la conversió entre A i B. On definirem la conversió d'atributs, per exemple entre A i B com:

$$C_{AB}$$

I definirem l'atribut X d'A com:

$$A_x$$

Com dèiem, per tal de computar la  $C_{AB}$  farem que, per cada relació entre un atribut d'A  $A_x$  i un atribut de B  $B_x$ , s'agafi el valor màxim entre tots els valors que ens donen tots els caminis tramitats per la relació  $(A_x - B_x)$ . D'aquesta manera obtenim la funció de conversió entre A i B.

#### 4.2.2 Mòdul: Xarxes privades

Les xarxes privades, o Xarxes de conversió privada, són aquelles les quals estan basades en dos assumpcions de privacitat. Primer, la existència de una relació que permet la conversió d'atributs entre dos dominis és només coneguda pels dominis que formen part de la relació, els quals seran anomenats veïns entre ells respectivament. La segona premissa de privacitat és que la funció de conversió entre els atributs d'A i de B només serà coneguda per el domini destí. Això significa que A i B coneixen l'existència de  $C_{AB}$  però només B coneix el contingut.

Dins les xarxes privades utilitzem una proposta col·laborativa per a la conversió d'atributs. La conversió s'aconsegueix mitjançant la propagació de les composicions intermitges, del càlcul d'atributs, a través de la xarxa de nodes. D'igual manera que a les xarxes públiques, es tracta de trobar la funció de conversió entre un domini origen i un domini destí. La base del funcionament de les xarxes privades es basa en que no es fa una cerca de camins dins del graf de dominis. La metodologia és més senzilla, els nodes envien el missatge a tots els seus veïns, és a dir, un domini envia missatge a tots els dominis amb els que té associada una conversió d'atributs. El procés del càlcul d'una funció de conversió entre un domini origen i domini destí determinats s'inicia indicant-ho al domini origen.

Quan un domini rep un missatge, el que fa és compondre el contingut de la relació provisional amb la seva corresponent funció de conversió, obtenint d'aquesta manera una nova funció de conversió provisional. Si el domini en qüestió no és el domini destí final, aquest enviara als seus veïns la funció de conversió provisional, això si, decrementant en un el valor de TTL del missatge. Si ens trobem en el cas de que hem arribat al domini destí definitiu, llavors aquest domini farà al mateix

que la resta, però en comptes d'enviar el missatge als seus veïns es quedara el valor final de la funció de conversió. Cal remarcar que un domini B conté les funcions de conversió dels dominis associats a ell, si associem A i D amb B, llavors B tindrà el contingut de  $C_{AB}$  i  $C_{DB}$ . És important remarcar això, donat que quan el domini B rebí un missatge haurà de comprovar qui li envia, (en aquest cas si A o D), per tal d'aplicar la funció de conversió correcta.

Tornant al tema del TTL és important dir que els missatges no estaran constantment donant voltes entre els dominis, cada cop que un domini rep un missatge, comprova el seu valor de TTL i si aquest és igual a 0, el missatge es descarta. Això és possible donat que com hem dit abans, cada cop que s'envia un missatge, es decrementa en un el seu valor de TTL. El TTL d'un missatge assegura la premissa de que els missatges no estan constantment donant voltes a la xarxa. També aquí es tindran en compte les bucles dins el recorregut de un missatge. Cada cop que un missatge arriba un domini s'afegeix al missatge dins la llista de nodes recorreguts. Al final de tot tindrem una llista amb tots els dominis recorreguts per el missatge, el que es fa és descartar les funcions de conversió extrems de missatges que han realitzat bucles en el seu viatge.

Un altre aspecte a destacar dins del comportament que tenen els dominis és el fet d'haver definit un timeout T. Significa que un domini X espera un temps T per rebre possibles missatges. Aquest temps es configura per tal d'assegurar que X rep totes les possibles relacions intermitges de tots els camins possibles, però sense fer que esperi eternament. Un domini X no sabrà com és la topologia de la xarxa, per tant serà impossible per X saber quants camins n'hi han i quantes funcions de conversió intermitges rebrà. Això fa que el procediment sigui no determinista, però el fa més flexible i eficient dins del punt de vista pràctic.

Tal com s'ha explicat al finalitzar la part de xarxes públiques, els missatges es van enviant de domini en domini fins que aquests arriben al domini destí. El domini destí agafarà la matriu de conversió que inclou cada missatge rebut i un

cop ha finalitzat el timeout d'espera realitza el càlcul de la funció de conversió definitiva. De la mateixa manera que es realitza per les xarxes públiques. Per acabar disposem de la figura 4.7 que conté el diagrama d'activitat d'un domini dins una xarxa privada.

## 4.3 Let's draw: MainDrawClass

Ens centrarem en aquest punt amb la part de l'entorn gràfic del nostre simulador. És important parlar sobre els mòduls d'estructuració d'informació i sobre els mòduls que ens aporten funcionalitat al nostre programa, donat que sense ells el projecte en sí no seria possible. També és important fer una ullada a tot el mòdul que ens a aportat interactivitat amb tota la estructura interna del programa. Ens referim al mòdul de l'entorn gràfic, la secció del programa que ens permet a través del ratolí definir gràfica i senzillament totes les relacions entre dominis i el contingut d'aquestes relacions.

### 4.3.1 Entorn de dibuix

En un primer moment, el que necessitem és un pla dins del espai on poder dibuixar. Necessitem un lloc dins el nostre programa on es puguin representar gràficament els nostres dominis. Es crea gràcies a la interfície gràfica de Java, Swing. Un pla que comença des del punt (0,0), a la cantonada superior esquerra de la nostra finestra de dibuix. A partir d'aquí, es defineix que quan es creï un domini dins la estructura interna del programa, es dibuixi una icona que representi el domini creat.

El punt interessant és que podem tenir els nostres dominis dibuixats, però ens sembla més interessant poder interactuar amb ells, poder moure'ls, poder definir propietats, etc. Aquí és on es crea el fet de poder agafar i arrossegat els objectes. Per tal d'aconseguir-ho ens guardem la posició de les icones en coordenades X i

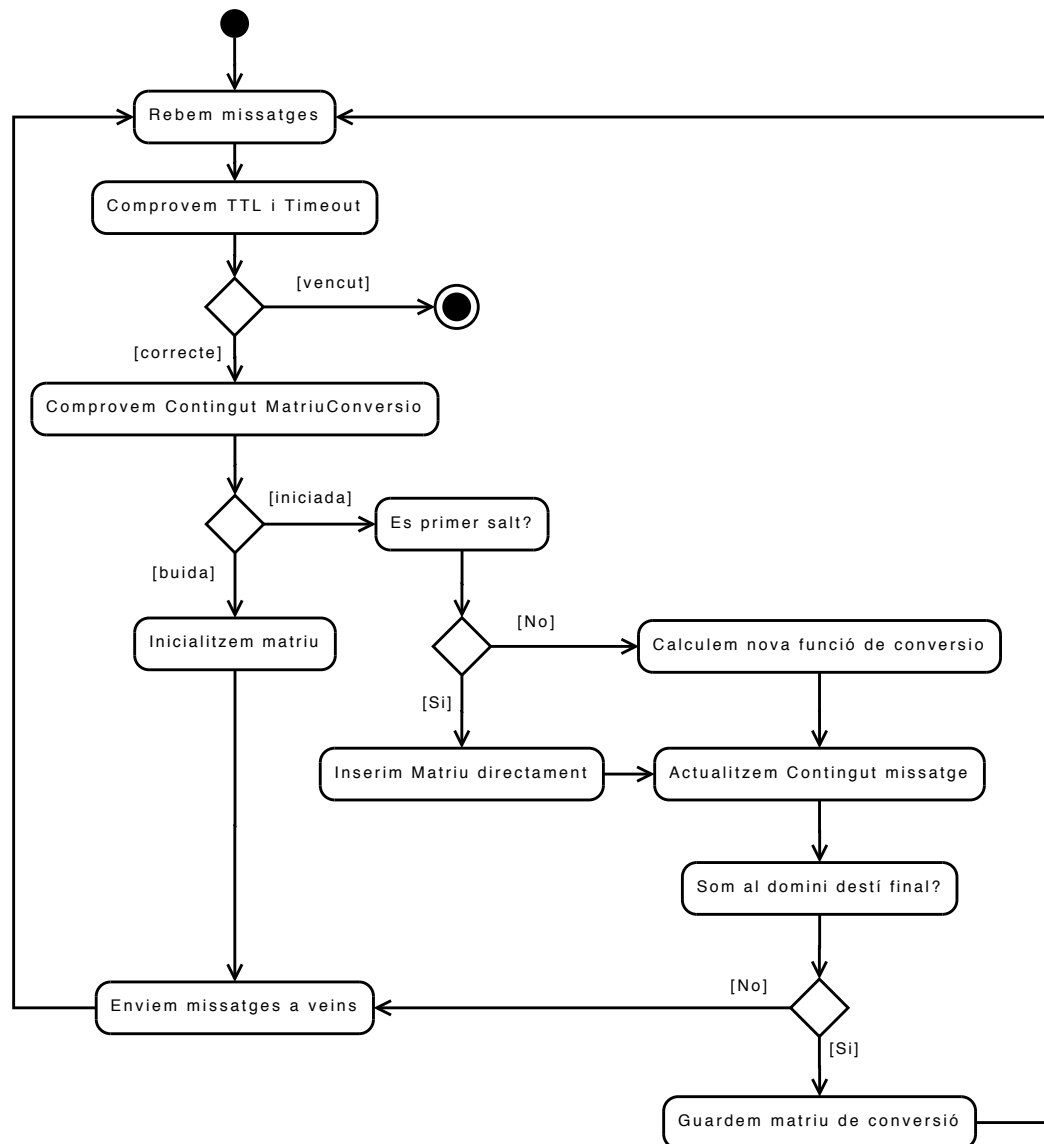


Figura 4.7: Diagrama d'activitat

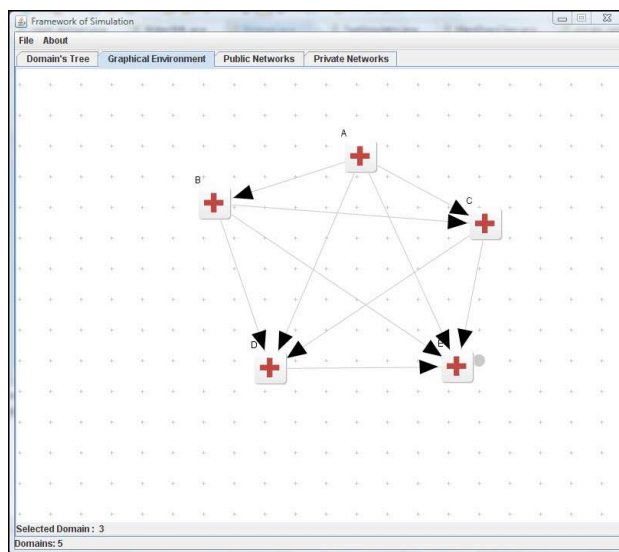


Figura 4.8: Captura de pantalla

Y dins del pla. Definim esdeveniments sobre el ratolí, per tal de captar quan es pressiona el boto esquerra o dret i quan s'allibera la pressió. D'aquesta manera i tramitant les distancies sobre el pla dependent dels esdeveniments realitzats, podem obtenir l'efecte d'agafar i arrossegar els objectes, fent que per cada moviment del ratolí l'escenari es torna a re-dibuixar. Tenim en compte que, si al pitjar el botó les coordenades obtingudes estan sobre el rang de una de les icones dels nostres dominis sabrem si hem fet clic sobre un domini, si no ha sigut així sabrem que em pitjat sobre el fons del escenari, permetent d'aquesta manera poder agafar i arrossegar el fons del escenari, definit una translació global de tot l'escenari sobre el pla (X,Y). D'aquesta manera disposem un pla infinit per a representar el nostre graf de dominis. Podem veure unes captures de pantalla a les figures 4.7 i 4.8 per poder apreciar el punt que acabem de comentar.

Un cop ja podem interactuar envers la movibilitat del escenari i dels elements que conté, donem un pas més enllà per tal de donar més funcionalitat al simulador. Aprofitem la estructura d'esdeveniments implementada per el ratolí i així podrem controlar quan fem clic sobre el botó dret sobre alguna de les icones dels

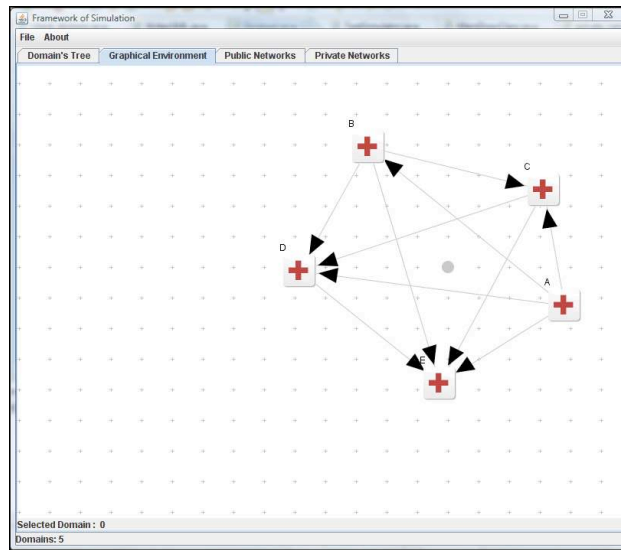


Figura 4.9: Captura de pantalla

nostres elements. Un cop fem clic dret, identificarem el domini seleccionat i a partir d'aquí mostrarem en una finestra secundària una llista on es donarà l'opció de crear una funció de conversió amb la resta de nodes. Sobre aquest punt, dins la interactivitat, en la finestra secundària també afegirem una llista, on ens donarà l'oportunitat d'eliminar totes les relacions de conversió establertes per el respectiu domini que tenim seleccionat.

Són tots aquests aspectes, dins l'entorn gràfic, els que ens permeten d'una forma clara i senzilla poder configurar tot l'entorn de dominis. Són petites eines que ens ajuden dins del marc de la interactivitat per tal d'obtenir una interoperabilitat sobre l'entorn multi-domini dins del nostre escenari gràfic.



# Capítol 5

## Proves i rendiment

Per aquest capítol s’han seleccionat i realitzat un conjunt de simulacions amb l’objectiu de valorar la capacitat del programa i els resultats obtinguts. En total s’han planificat de manera acurada més de 200 proves sobre el simulador. 200 d’aquestes queden plasmades directament sobre el document. La fiabilitat del programa ha estat validada amb l’estricta conjunt de comprovacions i contrastada a través dels resultats. Cal comentar que les proves han estat totes realitzades sobre un equip equipat amb un processador de doble nucli, on cada nucli funciona a 2,00Ghz. La màquina també disposa de 2,00 GB de RAM, on 256 MB són exclusius per a la tarja gràfica. És important remarcar que, donada la càrrega de treball que s’aplica en alguns moments sobre el simulador és necessari reservar 512 MB de RAM per a la JVM (Java Virtual Machine).

### 5.1 Resultats

Abans de començar, seria interessant puntualitzar que les composicions que realitzen els escenaris de conversió han estat comprovats prèviament, podem assegurar que tant els càlculs de matrius i els càlculs de camins es realitzen perfectament. Això ha estat validat detingudament durant el desenvolupament de l’aplicació amb petits exemples programats, donat que ara validar a mà un exemple de 20 dominis per 20 atributs seria impossible.

Dins la present secció realitzarem tot un conjunt de proves definides en tres grups. Farem un primer conjunt de proves per a entorns multi-domini sota l'escenari de conversió de xarxes públiques, després farem el mateix per a entorns multi-domini sota l'escenari de conversió de xarxes privades, la diferència per aquest és que definirem entorns amb i sense cicles en el recorregut entre dominis. Farem això per comprovar el cost que comporta realitzar càlculs quan es simulen entorns amb bucles. Cal comentar que no simular amb o sense cicles en un entorn on l'escenari de conversió són les xarxes públiques, donat que les xarxes públiques eliminen els camins amb cicles; L'única cosa que caldria puntualitzar és que el temps que triga en calcular els camins seria més alt.

Els paràmetres de simulació són els mateixos per els tres conjunts. Simularem de base des de per 5, 10, 15, 20 i 25 dominis i realitzarem les proves per 1 atribut per domini, fins a 20 atributs per domini amb intervals de 5 en 5. Tindrem simulacions per 1, 5, 10, 15 i 20 atributs. Cal comentar que aquestes simulacions són independents, el que volem dir és que les conversions fetes entre dominis són diferents per cada simulació de dominis amb diferents atributs, és a dir, les conversions fetes per la simulació de 20 dominis amb 5 atributs per domini són diferents per la simulació de 20 dominis amb 10 atributs o per la simulació de 10 dominis amb 1, 5, 10, 15 o 20 atributs. També és necessari afegir que pel que fa a les simulacions entre xarxes públiques o privades tindran el mateix escenari. Per exemple, una simulació feta amb 20 dominis i 15 atributs per domini té el mateix escenari definit per les públiques que per les privades.

### **5.1.1 Resultats - Xarxes Públiques**

Les gràfiques de resultats han estat obtingudes a partir de les simulacions en xarxes públiques. Veurem que tindrem l'eix Y que indica el número de matrius rebudes o el temps que es triga en calcular la funció de conversió, això depenent de la gràfica. L'eix X que indica el número de dominis. Les línies estan associades al número d'atributs per domini per les diferents simulacions. Cal clarificar que per

les xarxes públiques, el número de matrius rebudes és igual al número de camins que existeixen entre els dos dominis seleccionats. Això és obvi, donat que calcularem tots els camins entre les dues matrius, descartant els bucles i per cada camí obtindrem una matriu. Per les xarxes públiques serà el mateix parlar del número de matrius obtingudes o número de camins trobats.

Les dos gràfiques obtingudes són la figura 5.1 i la figura 5.2. La figura 5.1 ens mostra la progressió del número de matrius rebudes per els diferents entorns de simulació. La figura 5.2 ens mostra la mateixa progressió però en temps de computació. Respecte la figura 5.1 podem apreciar com per les simulacions per 15 atributs, línia morada, el número de matrius baixa per un entorn de 20 dominis a 25 dominis, això pot estar provocat per una aguda diferència en els escenaris simulats o per un enrederiment en els càlculs, cosa que podríem indicar con soroll dins la computació.

Cal indicar que per a poder simular, cal que els dominis estiguin connectats entre ells amb unes matrius de conversió definides de forma directa. El número de funcions de conversió directes definits entre dominis ha estat el doble que el número de dominis que hi ha en l'entorn a simular. És a dir, per la simulació amb 5 dominis, haurem definit 10 matrius de conversió entre dominis dins del entorn de simulació.

### 5.1.2 Valoracions resultats xarxes públiques

Pel que hem pogut veure a les gràfiques per les conversions realitzades, a les xarxes públiques tenim diferents punts importants a destacar.

- El número de matrius rebudes puja d'una manera bastant considerable a partir de 20 dominis en un escenari. Com es pot veure a la figura 5.1 el creixe-

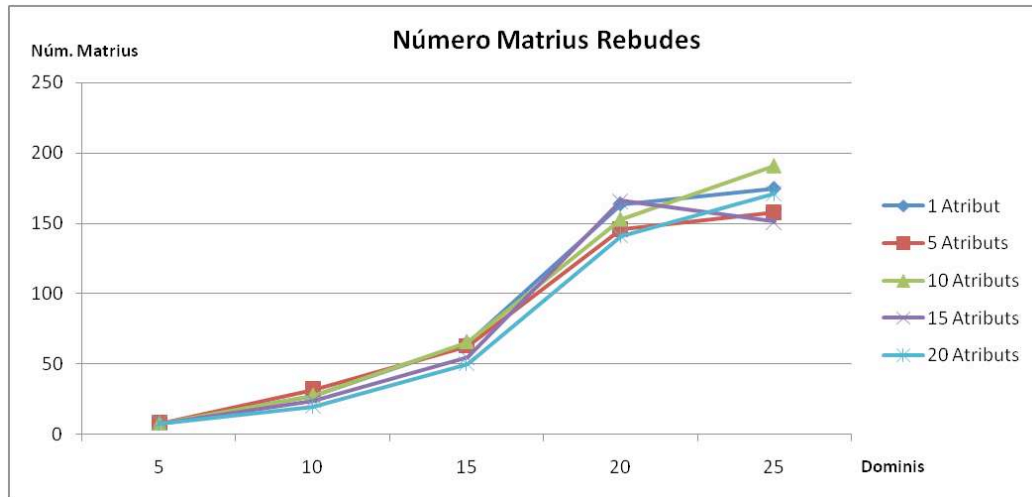


Figura 5.1: Xarxes públiques - Número matrius rebudes per diferents dominis

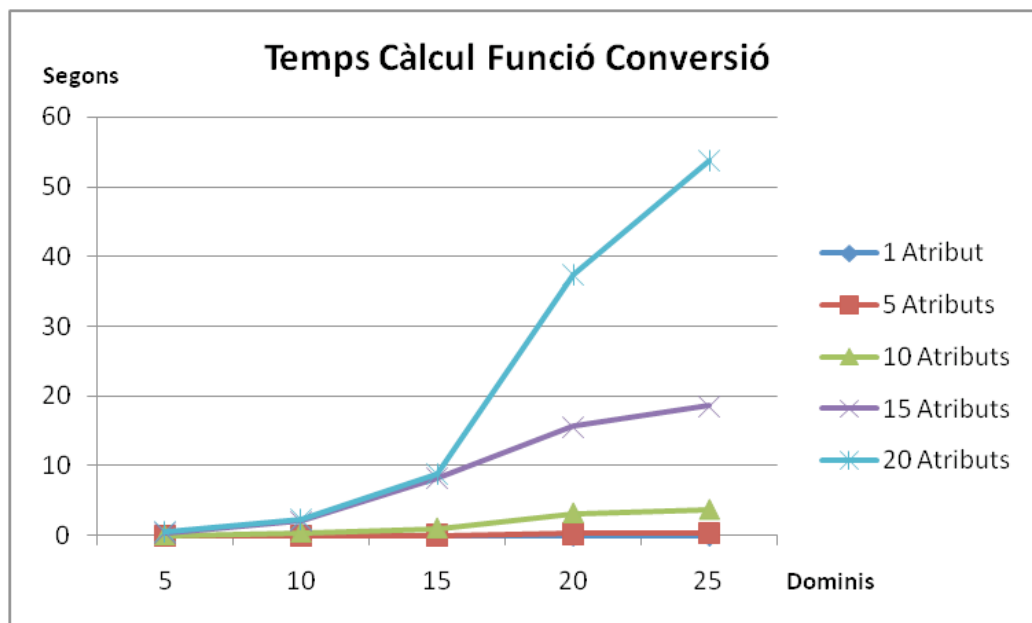


Figura 5.2: Xarxes públiques - Temps càlcul de matriu per diferents dominis

ment entre les simulacions per 15 a 20 dominis és de pràcticament el triple. La disposició de més dominis dins l'escenari dóna una major versatilitat dins l'escenari que proporciona una major quantitat de camins disponibles.

- Respecte al cost computacional podem veure a la figura 5.2 es fa més costos quan tenim més atributs. El temps que triga en calcular la funció de conversió és lineal per els diferents escenaris de dominis amb una quantitat petita d'atributs, però per quantitats d'atributs grans com 15 o 20, el creixement en temps de càlcul és exponencial. L'increment en el cost computacional bé donat per l'increment de càlculs i comparacions que haurem de fer segons anem tenint més atributs.
- Podem també apreciar a la figura 5.1, que la diferència del número de matrius que podem rebre per una simulació dins un entorn amb X dominis és més gran quan més dominis tinguem, és a dir, veiem per la simulació amb 5 dominis, que sempre hem tingut 8 matrius, en canvi, quant més dominis tenim en la simulació més diferència hi ha entre el número de matrius que podem rebre. Això és per la versatilitat de camins que ens aporta tindre més dominis a l'escenari. Una manera fàcil de veure-ho a la gràfica és mirant com per les diferents simulacions amb diferents dominis la diferència vertical és cada cop més gran.

### 5.1.3 Resultats - Xarxes privades sense cicles

Mostrem a continuació la gràfica per les xarxes privades sense cicles. La configuració és la mateixa que en el punt anterior. Tindrem l'eix Y que indica el número de matrius rebudes i l'eix X que indica el número de dominis. Les línies estan associades al número d'atributs per domini per les diferents simulacions. També hem indicat per cada línia el Timeout assignat per les simulacions. El valor de Timeout només ha variat depenent del número d'atributs definits per domini. El TTL és un valor fixe per totes les simulacions.

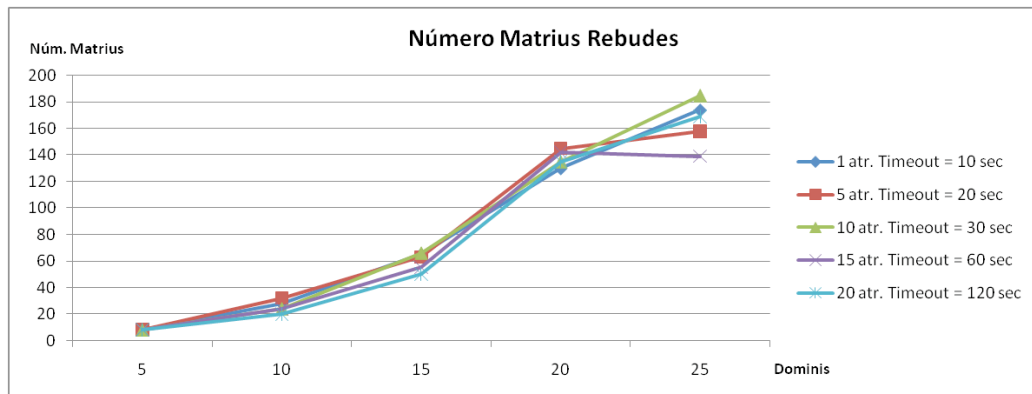


Figura 5.3: Xarxes privades - Número matriu rebudes per diferents dominis

### 5.1.4 Valoracions resultats xarxes privades sense cicles

Pel que hem pogut veure a la figura 5.3 per les conversions realitzades, a les xarxes públiques tenim diferents punts importants a destacar.

- Comparant la figura 5.1 i la figura 5.3 veiem que en la majoria de casos rebem menys matrius en les simulacions fetes per xarxes privades, que en xarxes públiques, això és degut a dos motius, el timeout i el TTL, aquí definim el temps que tenim obert el servidor per rebre peticions i el número de salts que pot fer un missatge per la xarxa, si aquests valors no són prou alts no ens arribaran els missatges i això vol dir que no tindrem totes les matrius que voldríem. Quant més alt sigui el número de dominis dins l'entorn a simular més en interessa incrementar aquests valors.
- La progressió del número de matrius rebudes és més lineal que en les xarxes públiques. En aquesta figura 5.3 també es pot veure el creixement entre els entorns de 15 a 20 dominis, però no arriba a ser tan gran. Això també és degut al motiu que acabem de comentar. El fet de tindre un entorn més gran, fa més probable que moltes matrius no tinguin temps suficient d'arribar al destí, ja sigui per falta de temps o falta de TTL.
- Per la gràfica podem apreciar aquí també que quant més atributs tenim més lent és l'entorn en general. Abans hem comentat el temps que trigava en

fer el calcul final de totes les matrius rebudes. Aquí podem apreciar que triga la primera matriu en arribar, quant més atributs per domini tenim més triguem en obtenir la matriu. Això es degut a que quan més atributs tenim, més comparacions i més càlculs tindrem que realitzar per tal d'obtenir els resultats segons es va movent el missatge per la xarxa.

### 5.1.5 Resultats - Xarxes privades amb cicles

Passem a continuació al tercer conjunt de proves. Ens trobem ara amb la simulació de xarxes privades però aquesta vegades amb cicles dins el nostre escenari. Utilitzarem aquí també entorns de 5, 10, 15, 20 i 25 dominis per 1, 5, 10, 15 i 20 atributs per domini. Definirem un número de relacions entre dominis que serà el doble que el número de dominis, i definirem un número de relacions per provocar cicles que serà la meitat del número de dominis que hi ha a l'entorn a simular. En aquest cas el timeout només varia per el nombre de dominis dins l'escenari i es realitzen diferents simulacions amb diferents valors del TTL per tots els escenaris. Això es degut a que ara, amb entorns amb bucles, ens interessa més simular amb un valor variable del TTL per els diferents escenaris i determinar el valor del timeout per el número de dominis que hi hagin a l'entorn. Fixarem el valor del timeout amb 15 segons pels entorns de 5 dominis, 20 segons per 10 dominis, 30 segons per 15 dominis i així duplicant el valor fins arribar 120 segons per 25 dominis.

Mostrem les gràfiques obtingudes, on la figura 5.4 ens mostra tot el contingut de totes les proves realitzades per les xarxes privades amb cicles. Potser el contingut d'informació és molt alt, però el que interessa veure és com el número de matrius rebudes en general decreix segons incrementem el número de dominis a l'entorn a simular. És interessant apreciar com el número de matrius rebudes també decreix segons incrementem el número d'atributs per domini. En aquest cas la diferència no és tan gran com per el decreixement de dominis per entorn. A la figura 5.5 es mostra l'evolució de les simulacions per els diferents atributs que poden definir per domini, focalitzant les proves en entorns amb només 15 domi-

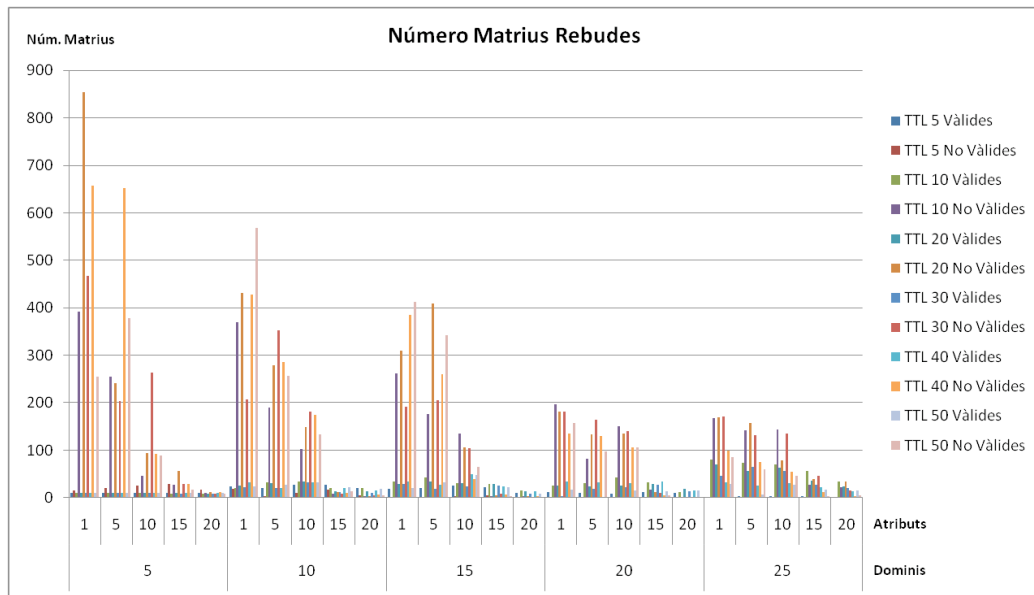


Figura 5.4: Xarxes privades amb bucles

Decreix el número de matrius rebudes segons incrementem el número de dominis i també segons augmenta el número d'atributs per domini

nis. Aquí es pot apreciar més detalladament com el número de matrius rebudes va decreixent segons augmentem el número d'atributs. Per la figura 5.6 es detalla el número de matrius rebudes per les diferents simulacions fixades amb un TTL de 20. Aquí també es pot apreciar com el número de matrius decreix segons augmentem el número dominis i d'atributs a l'entorn. Aquestes dues figures, 5.5 i 5.6, són per reflectir d'una manera detallada les apreciacions fetes a la figura 5.4.

### 5.1.6 Valoracions resultats xarxes privades amb cicles

Pel que hem pogut veure a les gràfiques, per aquestes conversions a les xarxes privades amb cicles tenim diferents punts importants a destacar.



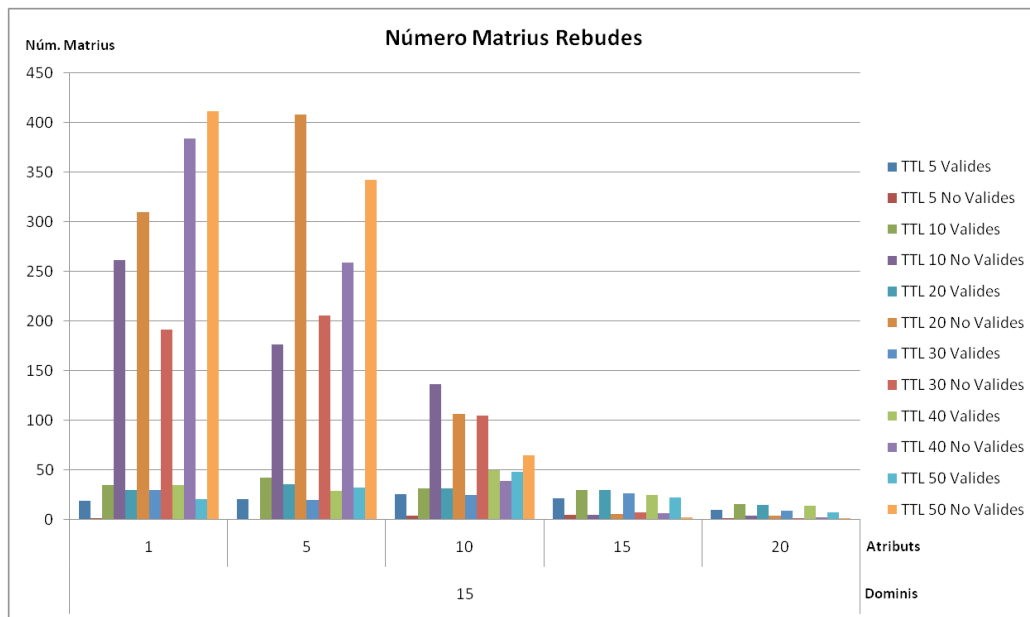


Figura 5.5: Xarxes privades amb bucles - Dades entorns amb 15 dominis

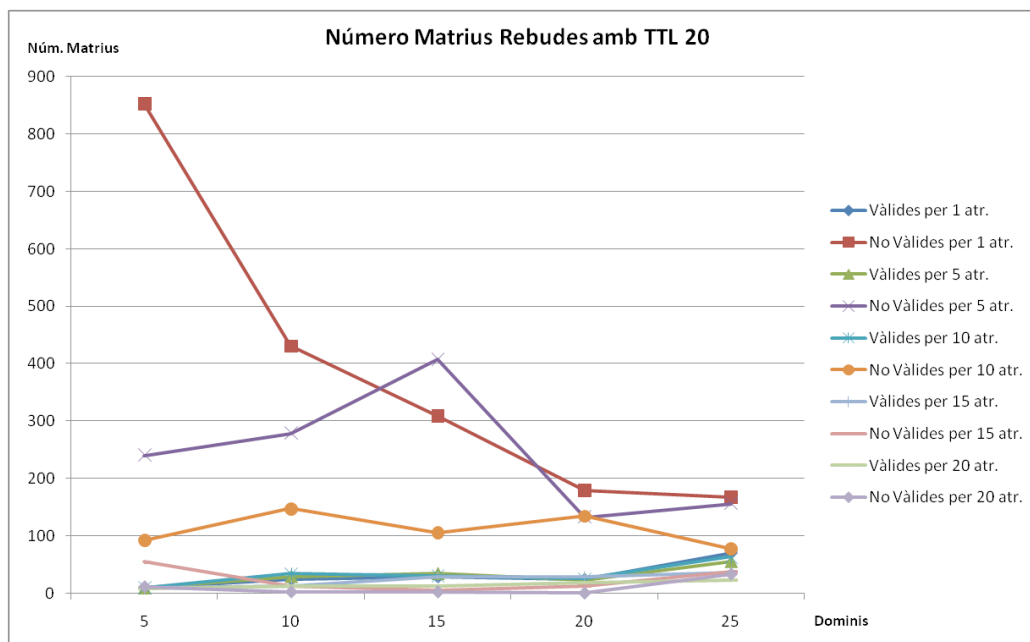


Figura 5.6: Xarxes privades amb bucles - Dades entorns amb TTL 20

- En un primer moment, comparant les figures 5.3 i 5.4, podem observar clarament que, el número de matrius vàlides, que són les ens interessen, és similar a les que podríem obtindre si no tinguéssim cicles, ara en per entorns a partir de 15 dominis, com es pot apreciar per la figura 5.5, el valor no és que es mantingui, sinó que es decrementa, fins arribar a valors molt baixos. Això ve donat perquè, per entorns on comencem a tindre un valor gran de dominis, tenim més camins amb bucles i més possibilitats de no arribar al destí.
- Amb la figura 5.4 veiem que les matrius no vàlides es disparen i normalment agafen el seu punt més alt per TTL igual a 20. A partir d'aquí el número és alt respecte a les vàlides però no tant com per TTL igual a 20. L'increment del TTL provoca un increment del número de missatges que recorreran la xarxa i aquest TTL tan gran pot provocar que puguin recórrer més, el fet de no tindre un timeout suficientment alt, farà que molts missatges es perdin. Aquesta situació no és tan accentuada segons s'incrementem el número de dominis a simular dins l'escenari, això es degut a que la xarxa per on pot viatjar un missatge es fa més gran.
- Es pot veure clarament, gràcies a la figura 5.5, que l'increment d'atributs per les simulacions fa decrementar el número de matrius rebudes, el cost que suposa incorporar més atributs dins una matriu comporta un increment en el cost computacional que provoca aquest efecte. Com s'ha comentat abans, més atributs significa més càlculs i comparacions per obtenir la funció de conversió. També afecta el fet de no tindre un timeout suficientment alt que farà que molts missatges no arribin al destí.
- Per la figura 5.6 es pot apreciar de forma clara que el número de matrius rebudes cau segons tenim més atributs per domini. També el número de matrius decreix segons augmentem el número de dominis a l'entorn. És important destacar dins aquesta figura que no hi ha un patró comú. Veiem, per 15 dominis i 5 atributs per domini tenim un increment en el número de matrius rebudes, això és perquè l'increment de dominis dins l'escenari

augmenta el número de camins amb bucles dins l'entorn i si afegim que hi han pocs atributs per domini, tindrem que el número de matrius rebudes s'incrementa fàcilment. Es pot apreciar també un curiós efecte on el número de matrius rebudes per 25 dominis és més alt que per 20 dominis, això és degut al canvi significatiu del timeout, per 20 dominis el timeout és de 60 segons i per 25 dominis és de 120 segons. Aquest és un canvi molt gran el qual fa que, donat el gran tamany de l'escenari i gràcies a un valor de timeout molt gran, arribin més matrius de conversió al domini destí.

- Podem veure que les matrius vàlides són les primeres que arriben, això ho podem veure en les situacions extremes on tindrem molts dominis amb molts atributs que simularem per un TTL igual a 5, o per TTL igual a 50. Veiem que tenim més matrius vàlides que no vàlides.

## 5.2 Valoració resultats

Comentarem per aquest punt un conjunt de conclusions i valoracions sobre els resultats obtinguts envers els tres conjunts simulats.

### 5.2.1 Comparació entre xarxes privades i xarxes públiques

Farem una petita comparació entre les xarxes privades i les xarxes públiques a partir dels resultats obtinguts. En un primer moment, donades les nostres simulacions, podem apreciar que pot ser molt més fiable treballar amb xarxes públiques. Això és per dos motius, primer ens assegurem de tindre la totalitat de totes les matrius possibles per els camins possibles dins l'escenari. Segon, obtenim aquests resultats en el menor temps possible. És cert que per les presents simulacions no estem tenint en compte que el punt fort de les xarxes privades, és a dir, la privacitat del domini envers les funcions de conversions entre ells, però a vistes de la nostra simulació ens surt més rentable utilitzar les xarxes públiques.

En els punts anterior, gràcies a les gràfiques, s'ha apreciat que el número de

matrius rebudes no distava molt entre les xarxes públiques i les xarxes privades sense cicles. És necessari recalcar el temps necessari per obtenir les matrius, per exemple, per 10 dominis amb 5 atributs per domini, obtenim 32 matrius en 64 mili-segons per les xarxes públiques, les xarxes privades aconseguix també totes les matrius però necessitem definir un timeout de 20 segons per tal d'obtenir totes les dades, potser en aquest cas podríem disminuir el timeout, però no més de 256 mili-segons que és el que ha trigat en arribar la primera matriu. És una diferència significativa. El problema més gran arriba quan incrementem el número de dominis i d'atributs i el cost comença a ser molt elevat, això farà que el timeout s'hagi d'incrementar i a més és possible que no obtinguem totes les matrius possibles.

Continuem la comparació però ara incloent les xarxes privades amb cicles. Si la diferència de matrius rebudes entre les xarxes públiques i les privades sense cicles podria arribar a ser un problema envers la qualitat o maximització de la funció de conversió resultant, ara aquest problema amb les xarxes privades amb cicles s'incrementa, donat que ara és més difícil arribar al domini destí i moltes matrius no arriben o arriben amb bucles dins el seu recorregut, i aquestes ja no ens interessen. Això és així perquè les xarxes privades amb bucles són més fidels a la realitat, si ens basem en les premisses en les que ha de treballar les xarxes privades. Hem de definir un TTL i un timeout, sabem que hem de definir un TTL suficientment gran com per a que els missatges arribin al final, però un valor massa alt pot fer créixer molt l'activitat a la xarxa i omplir els canals de missatges. Un TTL gran fa que tinguem més matrius no vàlides, però dins aquesta situació limitada no ens importa rebre matrius no vàlides, el que ens interessa realment és que ens arribin matrius vàlides per tal de maximitzar la funció de conversió.

Després de realitzar aquestes valoracions ens plantegem la següent pregunta. Dins un entorn amb 1000 dominis, quin seria el TTL necessari que hauríem de definir? Quin seria el timeout? Calcular quin seria el TTL i el Timeout necessari comportaria tot un conjunt de simulacions per poder estimar quins serien els valors ideals. Per el TTL no seria simplement que el valor fos superior al número de

dominis, el valor del TTL és exponencial respecte el número de dominis. Podem veure a les figures 5.5 i 5.6 com el TTL s'ha d'anar augmentant per tal de que arribin matrius i aquest augment moltes vegades no és suficient com per rebre un número suficient de matrius vàlides. Això és així degut a que tindre tants dominis a l'escenari farà que arribar al final del camí sigui molt més complicat. Un altre valor a calcular serà el del timeout, poder valorar quant de temps hem de tindre obert l'acceptació de missatges per part del server, pot tenir la mateixa complexitat que amb el càlcul del TTL. Un cop tinguem aquests valors, hem de tindre per suposat que encara que ideals, no disposarem de la totalitat de matrius vàlides i això pot comportar que no obtinguem la matriu maximitzada. Totes aquestes valoracions ens confirmen que el cost pot ser bastant alt i després de tot no se'ns assegura la seva eficiència, es per això que si volem estalviar-nos problemes davant la simulacions d'aquest entorn, haurem d'utilitzar les xarxes públiques. També trigaran cert temps en realitzar els càlculs però ens evitem l'estimació prèvia de paràmetres i ens assegurem que obtindrem la funció de conversió maximitzada.

### 5.2.2 Valoració rendiment i resultats obtinguts

Si fem una valoració al marge de si estem al davant de xarxes públiques o privades, podem dir respecte al simulador i els resultats obtinguts que ha proporcionat un rendiment eficient i uns valors compatibles dins les estimacions. El simulador ha pogut aguantar la càrrega de treball i a proporcionat els resultats dins dels límits que s'esperaven. El rendiment ha estat positiu, sobretot si tenim en compte que la màquina sobre la que s'ha fet el test no és del tot potent. En molts casos el que ens interessa és obtenir el valor, la cara llegible dels resultats i no en quant de temps ho hem obtingut. La realització d'aquestes 200 proves ens han aportat tot un conjunt de resultats que confirmen el que esperàvem, a més ens han aportat més informació útil per a la continuació del estudi dels mecanismes de control d'accés i interoperabilitat en entorns multi-domini.

### 5.3 Valoració de la configuració per xarxes privades

Valorarem dins el present punt com ens influeixen, respecte a un entorn de la vida real, els valors configurables de les polítiques de conversió que tenim definides per el nostre simulador. És cert que no podrem fer aquesta valoració per les xarxes públiques donat que no ens donen cap opció de configuració, però si que no podrem fer per les xarxes privades.

#### 5.3.1 Xarxes privades amb TTL i timeout

Comentarem aquí com ens poden afectar aquests atributs, com els podem configurar o com ens interessin depenent de la situació en la que ens trobem. Això sempre parlant dins del marc d'un entorn real. Com ja hem exposat en punt anteriors el TTL va quedar definit com el valor que s'assigna als missatges que viatgen per les xarxes privades, a més ens indica el número de salts que aquest missatge podrà realitzar dins la xarxa. També vam definir el valor de timeout T, on el valor indicava el temps que els dominis estarien esperant peticions, un cop que el timeout s'esgotés el domini deixara de rebre peticions.

Tindrem en compte que el valor de timeout és molt interessant, pel que fa a una simulació dins d'un entorn real. Exposem el cas, nosaltres definim el valor amb un grau més alt o més baix, si volem que els dominis esperin més o menys temps a rebre peticions. Aquest valor es podria definir o predeterminar llavors, depenent de la urgència de temps de la consulta, és a dir, si tinguéssim molta pressa per obtenir un resultat i en un cas on el temps sigui un be escàs, llavors definirem un timeout molt reduït, per tal d'obtenir una conversió del nostre atribut respecte el domini desitjat en el menor temps possible. Considerant que li doni temps a arribar al destí. En canvi, si per una altra banda, volem obtenir una conversió del nostre d'atribut respecte al domini desitjat, però volem que aquest sigui el més alt possible, llavors haurem de definir un temps timeout bastant alt per tal de que li doni temps a rebre el major nombre de funcions de conversió possible. D'aquesta manera podrem veure si és possible obtenir d'alguna de les conversions el major

factor de conversió. Serà idoni per la majoria de casos, trobar un valor equitatiu del timeout, per obtenir una conversió del nostre atribut amb un factor alt i que no trigui molt de temps en obtenir-ho. Dins aquest punt, comentar que la configuració del TTL ens vindrà determinada per l'equilibri de dos interessos. Primer, que un missatge que surti del origen tingui el TTL suficient com per que arribi al domini destí. I segon, que el valor de TTL no sigui un valor excessiu, per tal de no tindre missatges donant voltes per la xarxa sense cap tipus de sentit. Trobar l'equilibri del TTL pot ser més senzill que trobar el del timeout, però també comporta una cerca, a més si tenim en compte que potser tindre un TTL molt just en limitarà les opcions de tindre una funció de conversió maximitzada.





# Capítol 6

## Conclusions

Des d'una etapa inicial del projecte es va clarificar i establir des d'on partia el nostre projecte i es van destacar les bases inicials, definint d'aquesta manera uns objectius clars sobre l'aplicació. Valorem a continuació els objectius importants afrontats durant el desenvolupament del projecte.

- **S'ha realitzat un anàlisi en profunditat de quines eren les necessitats del nostre projecte.** Per realitzar això s'ha fet un estudi detallat sobre el treball previ realitzat envers els conceptes i metodologies que nosaltres tindríem que implementar a la nostra aplicació.
- **S'ha realitzat un disseny, basat en l'anàlisi previ, en el qual es van establir totes les premisses sobre el nostre projecte.** S'han establert quines serien les limitacions, característiques i necessitats de cada un dels mòduls que formaran l'aplicació. S'han definit de forma clara els conceptes sobre els quals el nostre simulador ha de treballar.
- **S'ha implementat aquest disseny sobre una aplicació, una eina de simulació en entorns multi-domini.** La nostra aplicació s'ha realitzat basant-nos de forma estricta sobre el disseny plantejat definint els mòduls d'una forma acurada. A més de la estructura interna d'informació, tal com estava previst, s'ha inclòs dins del simulador una interfície gràfica per tal de dotar a la aplicació amb interactivitat amb l'usuari.

- **S'ha realitzat tot un conjunt de més de 200 proves sobre el simulador.**

Amb les proves hem obtingut resultats concloents sobre l'estudi del que forma part per tal de comprovar la seva robustesa davant les diferents situacions que haurà de tramitar, com aplicació que tracta la simulació com objectiu. S'ha realitzat també una valoració satisfactòria dels resultats obtinguts.

Després de valorar els punts acabats de comentar podem afirmar sense cap problema que els objectius s'han acomplert de forma satisfactòria. Tal i com s'ha explicat al inici del present capítol, cal remarcar que l'acompliment d'aquests punts ha estat gràcies a una bona planificació i clarificació dels objectius en els moments inicials del projecte. Uns objectius clars, planificats i motivadors han estat la clau del resultat.

Al capítol anterior ja es van poder apreciar algunes conclusions i valoracions sobre els resultats obtinguts. S'han realitzat tot un conjunt de proves que ens han aportat resultats molt útils. Els valors obtinguts sempre han estat dins les expectatives i ens serveixen per confirmar estimacions prèvies o per valorar, envers el futur, el camí a agafar respecte la investigació. En un primer moment podem confirmar com, dins les simulacions realitzades, les xarxes públiques han estat les que ens han donat una resposta més ràpida i més fiable. Això no vol dir que les xarxes privades no ho facin però hem pogut apreciar com, a vegades i sobretot amb escenaris amb cicles, si no es disposa d'un valor de timeout i TTL adient no rebrem totes les matrius. No rebre totes les matrius pot comportar no finalitzar la simulació amb la matriu de conversió maximitzada. Per la majoria de casos observats la funció de conversió rebuda és la mateixa per els dos escenaris de conversió, només a les privades quan hem rebut poques matrius vàlides la matriu obtinguda no és maximitzada com podria ser a les xarxes públiques. La diferència important és el temps.

És necessari veure que aplicar aquests escenaris a la vida real comporta canvis. Per les xarxes públiques seria necessari que una entitat externa fos la que controlés totes les relacions entre dominis, sapigués quins són els camins i controlés totes

les funcions de conversió. Tot això per tal de proporcionar el resultat adient quan els usuaris d'un domini X realitzessin peticions sobre un altre domini Y. Respecte a les xarxes privades no caldria aquesta entitat controladora però tindríem el problema de la possible saturació de la xarxa amb el que això comporta, com podria ser no rebre tot el número possible de matrius vàlides.

Podem concloure aquest punt confirmant els punts més rellevants que hem pogut apreciar donades les taules i les gràfiques.

- S'ha vist que el fet de tindre més atributs aporta molta més informació a transmetre però el cost computacional és molt més gran. Això és fàcilment apreciable per les xarxes públiques en el temps que triga en obtenir la funció de conversió, que creix exponencialment i amb les xarxes privades amb cicles, on a partir de 15 atributs per domini el número de matrius rebudes creix considerablement.
- S'ha confirmat un punt que ja es preveia abans de realitzar el simulador i és que la disposició de més dominis dins un escenari ens pot aportar una gran varietat de camins que farà que el número de matrius rebudes es vegi incrementat.
- Sobre les xarxes privades s'ha vist molt clarament que l'increment del TTL pot jugar un paper perjudicial envers el número de matrius que pot rebre i la saturació de la xarxa. Això es veu clarament accentuat quan s'incrementa el número de dominis dins l'escenari. Una barreja d'aquests elements s'ha pogut apreciar de forma clara a les taules, on es podia interpretar que el motiu d'això venia provocat per l'alt index del TTL que prolongava el viatge dels missatges per la xarxa i el fet que hi haguessin tans dominis que multiplicava aquest efecte.

Arribats al final ens toca analitzar, si la planificació establerta ha estat satisfactòria, si aquesta ha estat completament viable des de bon principi o s'ha hagut

d'adaptar i com ha respòs davant de canvis o inconvenients durant el desenvolupament.

El que podem contestar davant aquestes premisses inicials que ens introdueix l'apartat és que el plantejament inicial ha sigut completament satisfactori. Això ha estat el resultat d'un conjunt d'accions plantejades inicialment i un seguit de circumstàncies favorables envers el desenvolupament del treball. Un dels objectius principals ha estat tenir clar des de bon principi que és el que s'havia de fer i valorar les eines disponibles.

També cal comentar que el resultat òptim s'ha obtingut gràcies a una bona planificació inicial i un desenvolupament constant i satisfactori que ha permès finalitzar les etapes a vegades una mica abans del temps establert, això ens ha donat la oportunitat tenir cura del desenvolupament ja realitzat i ens ha donat la opció plantejar amb calma i amb deteniment com havíem de tractar la següent etapa.

Cal comentar també que durant el desenvolupament del treball i de les diferents tasques van sorgir problemes que es tenien que resoldre. Aquests canvis i inconvenients no van suposar una problemàtica molt gran per diferents motius, com per exemple, en un primer moment, a l'hora de definir la planificació vam deixar clar que necessitaríem afegir temps per a possibles imprevistos com aquests, fet que ens donaria l'opció de poder mitigar el problema sense que afectés a la nostra planificació temporal. Un segon motiu ha estat la ràpida i constant comunicació entre coordinador i projectista que ha permès que davant l'aparició d'un problema es pogués parlar i prendre mesures ràpidament. Per tercer i últim sabem que des de bon principi necessitàvem que tots els conceptes relacionats amb el que volíem desenvolupar estiguessin ben clars, per tal de no desenvolupar una cosa que no era correcta o no estava basada en el que es volia.

Disposem a continuació un conjunt d'idees per tal de plantejar unes línies d'ampliació del treball realitzat. Seria interessant indagar una mica més en tot el

que el nostre projecte ens ha aportat i donar unes quantes idees per inspirar aquest treball futur.

- Al capítol introductori es va fer una distinció en zones, que representen d'una manera ordenada i realista una situació d'emergència. Seria interessant per fer el nostre projecte més realista, realitzar un simulador envers aquesta distinció. L'objectiu seria poder classificar dins el nostre escenari de simulació les diferents zones i implementar la interacció entre elles. Es podria definir per les diferents zones un conjunt de paràmetres configurables i establir per el conjunt de zones diferents valors per els atributs de timeout i TTL.
- Una opció interessant seria canviar la implementació feta dins les xarxes privades del nostre simulador i no realitzar la simulació amb sockets, sinó que podria ser interessant dissenyar l'escenari de conversió envers els agents mòbils. Plantejar d'igual manera els dominis com servidors però ara l'element que viatjaria de domini en domini seria un agent mòbil. Es podria apropar d'aquesta manera el simulador a moltes altres branques de desenvolupament amb les que treballa el SeNDA.
- Incloem també aquí una implementació que es va desestimar inicialment quan es plantejava el present projecte i està relacionat amb el punt anterior. Es podria plantejar de simular l'escenari dins un entorn més tangible, on cada ordinador fos un domini. D'aquesta manera es podria realitzar el mateix que hem fet amb el nostre projecte però utilitzant un cert nombre de ordinadors com a entitats del nostre escenari. També així es podria utilitzar la xarxa com a medi de comunicacions i utilitzar agents mòbils per tal de comunicar les màquines entre elles.



# Bibliografia

- [1] UML Unified Modeling Language, juny 2005.  
<<http://www.uml.org/>>
- [2] Security of Networks and Distributed Applications. Universitat Autònoma de Barcelona.  
<<http://www.senda.uab.es/>>
- [3] Departament d'Enginyeria de la Informació i de les Comunicacions.  
<<http://deic.uab.es/>>
- [4] Extensible Markup Language (XML).  
<<http://www.w3.org/XML/>>
- [5] A. Martín-Campillo, "Medigs, un sistema segur basat en agents per al descobriment i obtenció de dades mèdiques distribuïdes. estudi i desenvolupament." Universitat Autònoma de Barcelona. Juny 2007.
- [6] S. Robles, J. Borrell, J. Basart, R. Martí, J. Pons, G. Navarro-Arribas, J. Ametller, M. de Toro, M. Colobran, J. García-Alfaro, J. Cucurull, and C. Garrigues, "Diseño e implementación de una solución global para el soporte de servicios críticos en situaciones de emergencia," Ministerio de Educación y Ciencia, 2006.
- [7] A. Martín-Campillo, R. Martí, S. Robles, and C. Garrigues, "Historial clínico distribuido y seguro para situaciones de emergencias," Actas de la X RECSI, 2008.

- [8] C. Martínez-García, G. Navarro-Arribas, J. Borrell and A. Martín-Campillo. "An access control scheme for multi-agent systems over multi-domain environments. In 7th International Conference on Practical Applications of Agents and Multi-Agent Systems." Springer, Springer Berlin / Heidelberg, 2009. ISSN: 1615-3871.
- [9] C. Martínez-García. Control d'accés sota situacions d'emergència: Revisió i disseny de mecanismes de flexibilitat i interoperabilitat. Tesis de Máster. Universitat Autònoma de Barcelona. Juliol 2008.
- [10] Y. Sun, P. Pan, and X. Meng, "An approach for trusted interoperation in a multidomain environment" in *Autonomic and Trusted Computing*, 2006, pp. 352-361.
- [11] Developer Resources for Java Technology.  
<<http://java.sun.org/>>
- [12] Eclipse IDE Home.  
<<http://www.eclipse.org/>>
- [13] J. de Estado, "Ley orgánica 15/1999, de 13 de diciembre, de protección de datos de carácter personal." BOE número 298. Diciembre 1999.



---

Firmat: Fco. Javier Llorente Palacio  
Bellaterra, juny de 2009

## **Resum**

El present projecte part d'un estudi sobre el tractament de situacions d'emergència, on es considera que una infraestructura de comunicació, sobre aquest tipus d'escenaris, ha d'implementar la interoperabilitat i el control d'accés, entre els diferents elements que hi participen, utilitzant la conversió d'atributs. Per tal de materialitzar aquest estudi s'ha realitzat un anàlisi de les necessitats de l'aplicació i un disseny detallat dels mòduls que el formen. S'ha implementant d'aquesta manera una eina de simulació per entorns multi-domini. Finalment s'han realitzat un conjunt de proves per comprovar la seva fiabilitat respecte l'estudi original.

## **Resumen**

El presente proyecto parte de un estudio sobre el tratamiento de situaciones de emergencia, donde se considera que una infraestructura de comunicación, sobre este tipo de escenarios, tiene que implementar la interoperabilidad y el control de acceso, entre los diferentes elementos que participan, utilizando la conversión de atributos. Con tal de materializar este estudio se ha realizado un análisis de las necesidades de la aplicación y un diseño detallado de los módulos que lo forman. Se ha implementado de esta manera una herramienta de simulación para entornos multi-dominio. Finalmente se han realizado un conjunto de pruebas para comprobar su fiabilidad respecto al estudio original.

## **Abstract**

The present project departs from a study on the treatment of emergency situations, where it says that an infrastructure of communication, on this type of scenes, has to implement the interoperability and the control of access, between the different elements that take part on it, using the conversion of attributes. So to materialize this study there has been realized an analysis of the needs of the application and a detailed design of the modules that they form it. A tool of simulation for environments multi-domain has been implemented. Finally a set of tests have been realized to verify his reliability with regard to the original study.